Diplomarbeit

Untersuchung von Sequenzduplikaten auf funktionelle Eigenschaften im Referenzgenom von Homo Sapiens unter Verwendung eines relationalen Datenbanksystems

Gerd Anders

Betreuer: Dr. C. Gröpl, Dr. S. Heymann, P. Rieger
Erstgutachter: Prof. J. C. Freytag, Ph. D.
Zweitgutachter: Dr. C. Gröpl

Berlin, 12. Januar 2005
Danksagung

Ich möchte mich bei

• Dr. Stephan Heymann
• Dr. Clemens Gröpl
• Peter Rieger

insbesondere für die fachliche Betreuung, bei

• Prof. Johann - Christoph Freytag, Ph. D.

für die mir zur Verfügung gestellten Ressourcen und bei

• Heinz Werner

für die technisch - administrative Hilfe bedanken, sowie bei all denen, die mich während der Entstehungszeit dieser Diplomarbeit begleitet und unterstützt haben.

Selbständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Arbeit selbständig angefertigt und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet habe.


Gerd Anders
Inhaltsverzeichnis

1 Einleitung und Aufgabenstellung .................................................. 10

2 Biologische Grundlagen und Formalisierungen .............................. 12
   2.1 Biologische Grundlagen ....................................................... 12
       2.1.1 Aufbau und Struktur der DNA ...................................... 12
       2.1.2 Charakterisierung und biologische Interpretation der verwendeten Sequenzklassen .......... 15
   2.2 Formalisierungen .............................................................. 17

3 Berechnung der Duplikate .......................................................... 21
   3.1 BLAST ........................................................................... 22
       3.1.1 Kurze Beschreibung der Funktionsweise ......................... 22
       3.1.2 Vorverarbeitung und Anwendung ..................................... 22
   3.2 VMATCH ........................................................................ 23
       3.2.1 Kurze Beschreibung der Funktionsweise ......................... 23
       3.2.2 Vorverarbeitung und Anwendung ..................................... 23
       3.2.3 Ausgabetransformation über C-Schnittstelle .................. 25
   3.3 Parameterdiskussion ............................................................. 27
   3.4 Nachuntersuchungen zur Repeat- und Palindromanzahl .............. 28
   3.5 Kapitelzusammenfassung ....................................................... 30

4 Logischer Entwurf der DB SYNTENY ............................................. 31
   4.1 Schemata und Schemanamen ............................................... 32
   4.2 Datenbanktabellen ............................................................. 33
       4.2.1 Administrationskomponente ........................................... 34
       4.2.2 Sequenzkomponente ..................................................... 40
       4.2.3 Duplikatkomponente ..................................................... 43
       4.2.4 Partiell verwendete Attribute ........................................ 46
       4.2.5 Ergebniskomponente ..................................................... 46
   4.3 Trigger ........................................................................... 46
   4.4 Kapitelzusammenfassung ....................................................... 47

5 Überlegungen zum physischen Datenbankentwurf ......................... 48
   5.1 Tabellenbereiche ............................................................... 48
       5.1.1 Daten ..................................................................... 49
       5.1.2 LOBs ..................................................................... 50
       5.1.3 Indizes ..................................................................... 51
   5.2 Speicherbedarf für Protokolldateien ....................................... 51

3
# Inhaltsverzeichnis

<table>
<thead>
<tr>
<th>Kapitel</th>
<th>Titel</th>
<th>Seiten</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3</td>
<td>Pufferpools</td>
<td>52</td>
</tr>
<tr>
<td>5.4</td>
<td>Multidimensionales Clustering (MDC)</td>
<td>53</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Richtlinien zur Dimensionsschlüsselbestimmung</td>
<td>54</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Konfiguration des SYN_TS_DATA - Tabellenbereichs</td>
<td>57</td>
</tr>
<tr>
<td>5.4.3</td>
<td>Bestimmung der Seitengröße</td>
<td>58</td>
</tr>
<tr>
<td>5.5</td>
<td>Dateneinspeisung</td>
<td>59</td>
</tr>
<tr>
<td>5.6</td>
<td>Kapitelzusammenfassung und Fazit</td>
<td>60</td>
</tr>
<tr>
<td>6</td>
<td>Datenkontrolle &amp; -aufbereitung</td>
<td>61</td>
</tr>
<tr>
<td>6.1</td>
<td>DNA - Sequenzen</td>
<td>61</td>
</tr>
<tr>
<td>6.1.1</td>
<td>Datenkontrolle und -aufbereitung</td>
<td>61</td>
</tr>
<tr>
<td>6.1.2</td>
<td>Datenauswertung</td>
<td>62</td>
</tr>
<tr>
<td>6.2</td>
<td>Duplikatergebnisse</td>
<td>63</td>
</tr>
<tr>
<td>6.2.1</td>
<td>Datenkontrolle</td>
<td>65</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Datenaufbereitung</td>
<td>66</td>
</tr>
<tr>
<td>6.3</td>
<td>Kapitelzusammenfassung und Fazit</td>
<td>81</td>
</tr>
<tr>
<td>7</td>
<td>Statistische &amp; funktionelle Duplikatdatenauswertung</td>
<td>83</td>
</tr>
<tr>
<td>7.1</td>
<td>Die DB HS_CORE und FED</td>
<td>83</td>
</tr>
<tr>
<td>7.2</td>
<td>Vorbereitungen zu den Untersuchungen</td>
<td>84</td>
</tr>
<tr>
<td>7.2.1</td>
<td>Bestimmung intragenischer Abschnitte</td>
<td>84</td>
</tr>
<tr>
<td>7.2.2</td>
<td>Definition und Bestimmung (inter-)genischer Abschnitte</td>
<td>85</td>
</tr>
<tr>
<td>7.3</td>
<td>Qualitative Einschätzung der Duplikatdaten</td>
<td>86</td>
</tr>
<tr>
<td>7.4</td>
<td>Beschreibung der Untersuchungsziele und Ergebnisauswertung</td>
<td>89</td>
</tr>
<tr>
<td>7.4.1</td>
<td>Quantitativ - topologische Analyse</td>
<td>89</td>
</tr>
<tr>
<td>7.4.2</td>
<td>Qualitativ - topologische Analyse</td>
<td>90</td>
</tr>
<tr>
<td>7.5</td>
<td>Kapitelzusammenfassung</td>
<td>92</td>
</tr>
<tr>
<td>8</td>
<td>Grafische Ausgabe der Duplikate</td>
<td>93</td>
</tr>
<tr>
<td>8.1</td>
<td>Aufbau und Beschreibung des Eingabeformulars</td>
<td>94</td>
</tr>
<tr>
<td>8.2</td>
<td>Das Common Gateway Interface und Perl</td>
<td>96</td>
</tr>
<tr>
<td>8.3</td>
<td>Datenbankzugriff</td>
<td>97</td>
</tr>
<tr>
<td>8.4</td>
<td>Graphentheoretische Grundlagen</td>
<td>98</td>
</tr>
<tr>
<td>8.5</td>
<td>Datenvorverarbeitung</td>
<td>100</td>
</tr>
<tr>
<td>8.6</td>
<td>Umsetzung im Datenbanksystem</td>
<td>103</td>
</tr>
<tr>
<td>8.7</td>
<td>Clusterprogramm zur Berechnung minimal spannender Bäume</td>
<td>105</td>
</tr>
<tr>
<td>8.8</td>
<td>Grafikerstellung mit gnuplot</td>
<td>111</td>
</tr>
<tr>
<td>9</td>
<td>Zusammenfassung und Ausblick</td>
<td>116</td>
</tr>
<tr>
<td>A</td>
<td>Verwendete Abkürzungen</td>
<td>117</td>
</tr>
<tr>
<td>Literaturverzeichnis</td>
<td></td>
<td>119</td>
</tr>
<tr>
<td>B</td>
<td>Tabellen mit Laufzeiten</td>
<td>124</td>
</tr>
<tr>
<td>B.1</td>
<td>BLAST (ENSEMBL - VERSION 18_34)</td>
<td>124</td>
</tr>
<tr>
<td>B.2</td>
<td>VMATCH (ENSEMBL - VERSION 18_34)</td>
<td>124</td>
</tr>
<tr>
<td>B.3</td>
<td>Datenbank / C - Programm(e)</td>
<td>127</td>
</tr>
</tbody>
</table>
C Tabellen mit Anzahl der Duplikate

C.1 Tabellen zu den unbestimmten DNA-Sequenzen
C.1.1 homo sapiens (unmasked) ........................................... 129
C.1.2 pan troglodytes (unmasked) ...................................... 130
C.1.3 homo sapiens (masked) ............................................. 130

C.2 Anzahl aller Duplikate .................................................. 131
C.2.1 homo sapiens (unmaskiert) ....................................... 131
C.2.2 mus musculus (unmaskiert) ...................................... 132
C.2.3 pan troglodytes (unmaskiert) .................................. 133
C.2.4 homo sapiens (maskiert) .......................................... 134

C.3 Anzahl redundanter und nichtredundanter Repeats und Palindrome ........................................... 135
C.3.1 homo sapiens (unmaskiert) ....................................... 135
C.3.2 pan troglodytes (unmaskiert) .................................. 136
C.3.3 homo sapiens (maskiert) .......................................... 137
C.3.4 homo sapiens (unmaskiert) ....................................... 138
C.3.5 pan troglodytes (unmaskiert) .................................. 139
C.3.6 homo sapiens (maskiert) .......................................... 140
C.4 native Clustering ....................................................... 140

D Abbildungen

D.1 Abbildungen zur durchschnittlichen Duplikatarmlänge in diversen Topologien des unmaskierten Humangenoms ........................................... 148
D.2 Abbildungen zur prozentualen Duplikatverteilung in diversen Topologien des unmaskierten Humangenoms ........................................... 152
<table>
<thead>
<tr>
<th>3.1 DNA - Längen der humanen Chromosomen in Bp (ENSEMBL V. 23)</th>
<th>21</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1 Die Datensätze der DBT admin.species</td>
<td>35</td>
</tr>
<tr>
<td>4.2 Auszug aus der DBT admin.version</td>
<td>35</td>
</tr>
<tr>
<td>4.3 Auszug aus der DBT admin.species_version</td>
<td>36</td>
</tr>
<tr>
<td>4.4 Die Datensätze der DBT admin.vmatch_seqtypes</td>
<td>36</td>
</tr>
<tr>
<td>4.5 Die Datensätze der DBT admin.vmatch_param_seqtypes</td>
<td>37</td>
</tr>
<tr>
<td>4.6 Die Datensätze der DBT admin.repeat_pid</td>
<td>37</td>
</tr>
<tr>
<td>4.7 Die Datensätze der DBT admin.palindrome_pid</td>
<td>38</td>
</tr>
<tr>
<td>4.8 Auszug aus der DBT admin.supernormal_repeat_id</td>
<td>38</td>
</tr>
<tr>
<td>4.9 Auszug aus der DBT admin.tandem_id</td>
<td>39</td>
</tr>
<tr>
<td>4.10 Auszug aus der DBT admin.dna_carrier</td>
<td>39</td>
</tr>
<tr>
<td>5.1 Aufteilung der DMS - Tabellenbereiche</td>
<td>49</td>
</tr>
<tr>
<td>5.2 geänderte Protokollierungsparameter</td>
<td>52</td>
</tr>
<tr>
<td>5.3 32 Bit Betriebssysteme und deren max. shared memory</td>
<td>52</td>
</tr>
<tr>
<td>5.4 Zusammensetzung des shared memory aus DB - Komponenten</td>
<td>53</td>
</tr>
<tr>
<td>5.5 db2advis - Ergebnisse für Abfrage 5.1</td>
<td>56</td>
</tr>
<tr>
<td>5.6 db2advis - Ergebnisse für Abfrage 5.2</td>
<td>56</td>
</tr>
<tr>
<td>5.7 db2advis - Ergebnisse nach Erzeugung der empfohlenen Indizes für Abfrage 5.2</td>
<td>57</td>
</tr>
<tr>
<td>5.8 Informationen zu den Tabellenbereichen vor dem Laden</td>
<td>58</td>
</tr>
<tr>
<td>5.9 Informationen zu den Tabellenbereichen vor dem Laden</td>
<td>58</td>
</tr>
<tr>
<td>5.10 Laufzeiten für load und set integrity</td>
<td>59</td>
</tr>
<tr>
<td>5.11 Laufzeiten für speziesweisen Sequenzimport</td>
<td>60</td>
</tr>
<tr>
<td>6.1 IUPAC - Konvention zur Codierung von Basen</td>
<td>62</td>
</tr>
<tr>
<td>6.2 Auszug aus der DBT duplicates.repeats (maskiertes Chr. 22 von homo sapiens)</td>
<td>69</td>
</tr>
<tr>
<td>6.3 Laufzeiten dia_value - Indexerstellung mit runstats</td>
<td>70</td>
</tr>
<tr>
<td>6.4 timernons - Ergebnisse und Speicherplatzbelegung für beide Indexes</td>
<td>70</td>
</tr>
<tr>
<td>6.5 Laufzeiten der Redundanzerkennung durch SQLPL - Prozedur</td>
<td>71</td>
</tr>
<tr>
<td>6.6 Laufzeiten der Redundanzerkennung durch C - Programm</td>
<td>71</td>
</tr>
<tr>
<td>6.7 Laufzeiten der Redundanzerkennung durch C - Programm</td>
<td>74</td>
</tr>
<tr>
<td>6.8 Auszug aus der DBT duplicates.repeats_native (maskiertes Chr. 22 von homo sapiens)</td>
<td>75</td>
</tr>
<tr>
<td>6.9 Laufzeiten zum Setzen der neuen Mismatchanzahl</td>
<td>78</td>
</tr>
<tr>
<td>6.10 Einige Laufzeiten zum Setzen der overlap_id - Attributwerte</td>
<td>81</td>
</tr>
<tr>
<td>7.1 Beispieldaten vom unmaskierten humanen Chromosom 1</td>
<td>87</td>
</tr>
</tbody>
</table>
# TABELLENVERZEICHNIS

<table>
<thead>
<tr>
<th>S.1</th>
<th>Kenndaten zur Indexerstellung</th>
<th>105</th>
</tr>
</thead>
<tbody>
<tr>
<td>B.1</td>
<td>Laufzeiten und proz. CPU - Auslastung BLAST - Vorverarbeitung</td>
<td>124</td>
</tr>
<tr>
<td>B.2</td>
<td>Laufzeiten und prozentuale CPU - Auslastung VMATCH - Vorverarbeitung</td>
<td>125</td>
</tr>
<tr>
<td>B.3</td>
<td>Anz. Tandems, VMATCH-Laufzeiten (Sek.) und Prozessorauslastung</td>
<td>125</td>
</tr>
<tr>
<td>B.4</td>
<td>Anz. supermax. Repeats, VMATCH-Laufzeiten (Sek.) und Prozessorauslastung</td>
<td>126</td>
</tr>
<tr>
<td>B.5</td>
<td>Anz. Repeats, VMATCH-Laufzeiten (Sek.) und Prozessorauslastung (I)</td>
<td>126</td>
</tr>
<tr>
<td>B.6</td>
<td>Anz. Repeats, VMATCH-Laufzeiten (Sek.) und Prozessorauslastung (II)</td>
<td>127</td>
</tr>
<tr>
<td>B.7</td>
<td>Anz. Palindrome, VMATCH-Laufzeiten (Sek.) und Prozessorauslastung (I)</td>
<td>127</td>
</tr>
<tr>
<td>B.8</td>
<td>Anz. Palindrome, VMATCH-Laufzeiten (Sek.) und Prozessorauslastung (II)</td>
<td>128</td>
</tr>
<tr>
<td>B.9</td>
<td>Laufzeiten zur Redundanzerkennung mit SQL PL</td>
<td>128</td>
</tr>
<tr>
<td>B.10</td>
<td>Laufzeiten zur Redundanzerkennung mit C - Programm</td>
<td>128</td>
</tr>
<tr>
<td>C.1</td>
<td>Ergebnisse zu unbestimmten Sequenzabschnitten</td>
<td>129</td>
</tr>
<tr>
<td>C.2</td>
<td>Ergebnisse zu unbestimmten Sequenzabschnitten</td>
<td>130</td>
</tr>
<tr>
<td>C.3</td>
<td>Ergebnisse zu unbestimmten Sequenzabschnitten</td>
<td>130</td>
</tr>
<tr>
<td>C.4</td>
<td>Anzahl Duplikate nach Klassen aufgeschlüsselt</td>
<td>131</td>
</tr>
<tr>
<td>C.5</td>
<td>Anzahl Duplikate nach Klassen aufgeschlüsselt</td>
<td>132</td>
</tr>
<tr>
<td>C.6</td>
<td>Anzahl Duplikate nach Klassen aufgeschlüsselt</td>
<td>133</td>
</tr>
<tr>
<td>C.7</td>
<td>Anzahl Duplikate nach Klassen aufgeschlüsselt</td>
<td>134</td>
</tr>
<tr>
<td>C.8</td>
<td>Anzahl redundant und nichtredundanter Repeats</td>
<td>135</td>
</tr>
<tr>
<td>C.9</td>
<td>Anzahl redundant und nichtredundanter Repeats</td>
<td>136</td>
</tr>
<tr>
<td>C.10</td>
<td>Anzahl redundant und nichtredundanter Repeats</td>
<td>137</td>
</tr>
<tr>
<td>C.11</td>
<td>Anzahl redundant und nichtredundanter Palindrome</td>
<td>138</td>
</tr>
<tr>
<td>C.12</td>
<td>Anzahl redundant und nichtredundanter Palindrome</td>
<td>139</td>
</tr>
<tr>
<td>C.13</td>
<td>Anzahl redundant und nichtredundanter Palindrome</td>
<td>140</td>
</tr>
<tr>
<td>C.14</td>
<td>Repeatstatistik zum native clustering von <em>homo sapiens</em> (unmaskiert)</td>
<td>141</td>
</tr>
<tr>
<td>C.15</td>
<td>Repeatstatistik zum native clustering von <em>pan troglodytes</em> (unmaskiert)</td>
<td>142</td>
</tr>
<tr>
<td>C.16</td>
<td>Repeatstatistik zum native clustering von <em>mus musculus</em> (unmaskiert)</td>
<td>143</td>
</tr>
<tr>
<td>C.17</td>
<td>Palindromstatistik zum native clustering von <em>homo sapiens</em> (ummaskiert)</td>
<td>144</td>
</tr>
<tr>
<td>C.18</td>
<td>Palindromstatistik zum native clustering von <em>pan troglodytes</em> (ummaskiert)</td>
<td>145</td>
</tr>
<tr>
<td>C.19</td>
<td>Palindromstatistik zum native clustering von <em>mus musculus</em> (ummaskiert)</td>
<td>146</td>
</tr>
</tbody>
</table>
Abbildungsverzeichnis

<table>
<thead>
<tr>
<th>Kapitel</th>
<th>Abbildung</th>
<th>Seite</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Cytosin</td>
<td>13</td>
</tr>
<tr>
<td>2.2</td>
<td>Thymin</td>
<td>13</td>
</tr>
<tr>
<td>2.3</td>
<td>Guanin</td>
<td>13</td>
</tr>
<tr>
<td>2.4</td>
<td>Adenin</td>
<td>13</td>
</tr>
<tr>
<td>2.5</td>
<td>Desoxyribose</td>
<td>14</td>
</tr>
<tr>
<td>2.6</td>
<td>Ribose</td>
<td>14</td>
</tr>
<tr>
<td>2.7</td>
<td>Phosphorsäure</td>
<td>14</td>
</tr>
<tr>
<td>2.8</td>
<td>Nukleotid</td>
<td>14</td>
</tr>
<tr>
<td>2.9</td>
<td>Die DNA-Doppelhelix mit komplementären Basenpaaren</td>
<td>15</td>
</tr>
<tr>
<td>2.10</td>
<td>Beispiel für ein Repeat</td>
<td>19</td>
</tr>
<tr>
<td>2.11</td>
<td>Beispiel für ein Palindrom</td>
<td>19</td>
</tr>
<tr>
<td>3.1</td>
<td>Anzahl berechneter Repeats und Palindrome im unmaskierten Humangenom</td>
<td>26</td>
</tr>
<tr>
<td>3.2</td>
<td>Anzahl berechneter Duplikate (ohne Tandems und supermax. Repeats)</td>
<td>29</td>
</tr>
<tr>
<td>4.1</td>
<td>Logische Datenbankzuordnung</td>
<td>32</td>
</tr>
<tr>
<td>4.2</td>
<td>Direkte logische Abhängigkeiten zwischen den Schemata</td>
<td>33</td>
</tr>
<tr>
<td>4.3</td>
<td>Datenbankschema der Administrationskomponente</td>
<td>34</td>
</tr>
<tr>
<td>4.4</td>
<td>Datenbankschema der Sequenzkomponente</td>
<td>40</td>
</tr>
<tr>
<td>4.5</td>
<td>Mustertrigger aus Abschnitt 4.3 von Seite 46</td>
<td>47</td>
</tr>
<tr>
<td>6.1</td>
<td>Schema der Verarbeitungskette der DNA-Sequenzen</td>
<td>61</td>
</tr>
<tr>
<td>6.2</td>
<td>Sequenzstatistik</td>
<td>63</td>
</tr>
<tr>
<td>6.3</td>
<td>Schema der Verarbeitungskette von Repeats und Palindromen</td>
<td>64</td>
</tr>
<tr>
<td>6.4</td>
<td>Unterschiedliche lange Palindromsequenzen mit gleichen Startpositionen</td>
<td>66</td>
</tr>
<tr>
<td>6.5</td>
<td>Repeat (V) mit (Sub-)Repeats (U, W)</td>
<td>67</td>
</tr>
<tr>
<td>6.6</td>
<td>2D-Darstellung von Abb. [6.3]</td>
<td>67</td>
</tr>
<tr>
<td>6.7</td>
<td>(Eingerahmtes) Palindrom mit eingetragenen Diagonalwerten</td>
<td>69</td>
</tr>
<tr>
<td>6.8</td>
<td>Zwei sich teilweise überlappende Repeats (E, F)</td>
<td>72</td>
</tr>
<tr>
<td>6.9</td>
<td>Vergleich der Repeatlängen mit und ohne native Clustering</td>
<td>76</td>
</tr>
<tr>
<td>6.10</td>
<td>Vergleich der Palindromlängen mit und ohne native Clustering</td>
<td>76</td>
</tr>
<tr>
<td>6.11</td>
<td>Native Repeatclusterstatistik</td>
<td>77</td>
</tr>
<tr>
<td>6.12</td>
<td>Native Palindromclusterstatistik</td>
<td>77</td>
</tr>
<tr>
<td>6.13</td>
<td>Mögliche Mismatchverteilung (‘x’) bei überlappenden Repeats</td>
<td>78</td>
</tr>
<tr>
<td>6.14</td>
<td>Mögliche Mismatchverteilung (‘x’) bei überlappenden Repeats</td>
<td>79</td>
</tr>
<tr>
<td>6.15</td>
<td>Anwendungsbeispiel zum extended native clustering</td>
<td>79</td>
</tr>
<tr>
<td>7.1</td>
<td>Schema der topologischen Ebenen</td>
<td>85</td>
</tr>
<tr>
<td>Abbildungsverzeichnis</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>7.2 Visualisierung des Anstiegs $\alpha_k$</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>7.3 Visualisierung von Daten aus Tabelle 7.1</td>
<td>88</td>
<td></td>
</tr>
<tr>
<td>7.4 Proz. Aufteilung der Palindrome in diverse Topologien im unmaskierten Humangenom</td>
<td>90</td>
<td></td>
</tr>
<tr>
<td>7.5 $\varnothing$ Palindromarmlängen in diversen Topologien im unmaskierten Humangenom</td>
<td>91</td>
<td></td>
</tr>
<tr>
<td>8.1 Schema der Verarbeitungskette zur grafischen Darstellung der Duplikate</td>
<td>93</td>
<td></td>
</tr>
<tr>
<td>8.2 Eingabeformular zur Duplikatvisualisierung</td>
<td>95</td>
<td></td>
</tr>
<tr>
<td>8.3 Kanteneinzeichnung im MST</td>
<td>99</td>
<td></td>
</tr>
<tr>
<td>8.4 Zugriffsplan vor Indexerstellung von 8.1 mit timerons</td>
<td>106</td>
<td></td>
</tr>
<tr>
<td>8.5 Zugriffsplan nach Indexerstellung von 8.1 mit timerons</td>
<td>106</td>
<td></td>
</tr>
<tr>
<td>8.6 Zugriffsplan vor Indexerstellung von 8.2 mit timerons</td>
<td>107</td>
<td></td>
</tr>
<tr>
<td>8.7 Zugriffsplan nach Indexerstellung von 8.2 mit timerons</td>
<td>107</td>
<td></td>
</tr>
<tr>
<td>8.8 Zugriffsplan vor Indexerstellung von 8.3 mit timerons</td>
<td>108</td>
<td></td>
</tr>
<tr>
<td>8.9 Zugriffsplan nach Indexerstellung von 8.3 mit timerons</td>
<td>108</td>
<td></td>
</tr>
<tr>
<td>8.10 Schema der Verarbeitungskette des Clusterprogramms</td>
<td>109</td>
<td></td>
</tr>
<tr>
<td>8.11 Schema der Duplikatliste</td>
<td>110</td>
<td></td>
</tr>
<tr>
<td>8.12 Schema der Duplikatliste</td>
<td>111</td>
<td></td>
</tr>
<tr>
<td>8.13 Grober Auszug vom humanen Chromosom Y</td>
<td>112</td>
<td></td>
</tr>
<tr>
<td>8.14 Verfeinerter Auszug von Abb. 8.13</td>
<td>114</td>
<td></td>
</tr>
<tr>
<td>8.15 Verfeinerter Auszug der unteren Diagonale von Abb. 8.14</td>
<td>114</td>
<td></td>
</tr>
<tr>
<td>8.16 Verfeinerter Auszug der oberen Diagonale von Abb. 8.14</td>
<td>115</td>
<td></td>
</tr>
<tr>
<td>D.1 Anzahl berechneter Repeats und Palindrome im Schimpansen genom</td>
<td>147</td>
<td></td>
</tr>
<tr>
<td>D.2 Anzahl berechneter Repeats und Palindrome im Mausgenom</td>
<td>148</td>
<td></td>
</tr>
<tr>
<td>D.3 $\varnothing$ Palindromarmlängen</td>
<td>149</td>
<td></td>
</tr>
<tr>
<td>D.4 $\varnothing$ native Palindromarmlängen</td>
<td>149</td>
<td></td>
</tr>
<tr>
<td>D.5 $\varnothing$ Palindromarmlängen</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>D.6 $\varnothing$ Repeatarmlängen</td>
<td>150</td>
<td></td>
</tr>
<tr>
<td>D.7 $\varnothing$ Repeatarmlängen</td>
<td>151</td>
<td></td>
</tr>
<tr>
<td>D.8 $\varnothing$ native Repeatarmlängen</td>
<td>151</td>
<td></td>
</tr>
<tr>
<td>D.9 $\varnothing$ native Repeatarmlängen</td>
<td>152</td>
<td></td>
</tr>
<tr>
<td>D.10 Proz. Aufteilung der Palindrome</td>
<td>153</td>
<td></td>
</tr>
<tr>
<td>D.11 Proz. Aufteilung der native Palindrome</td>
<td>153</td>
<td></td>
</tr>
<tr>
<td>D.12 Proz. Aufteilung der native Palindrome</td>
<td>154</td>
<td></td>
</tr>
<tr>
<td>D.13 Proz. Aufteilung der Repeats</td>
<td>154</td>
<td></td>
</tr>
<tr>
<td>D.14 Proz. Aufteilung der Repeats</td>
<td>155</td>
<td></td>
</tr>
<tr>
<td>D.15 Proz. Aufteilung der native Repeats</td>
<td>155</td>
<td></td>
</tr>
<tr>
<td>D.16 Proz. Aufteilung der native Repeats</td>
<td>156</td>
<td></td>
</tr>
</tbody>
</table>


In der Folgezeit wurden zu mehreren Humanchromosomen, u. a. im Februar 2003 zu Chromosom vierzehn ([HE03]), im Juni 2003 zu Chromosom Y ([SK03]), im Juli 2003 zu Chromosom sieben ([HF03]), im Oktober 2003 zu Chromosom sechs ([M03]), Artikel veröffentlicht. Trotz bisher erreichter Erfolge wird weiterhin mit Nachdruck an der Sequenzierung des Human- genoms gearbeitet, da sich mitunter ganze Chromosomenarme der Sequenzierung mit den bisherigen Methoden entzogen haben.


Diese Diplomarbeit verfolgt zwei Ziele: Das primäre Ziel besteht im Aufbau einer Verarbei-
tungskette, die mit der Berechnung von wiederkehrenden Sequenzabschnitten beginnt und in einer Plattformunabhängigen Visualisierung einschließlich diverser Filterungsmöglichkeiten endet. Das sekundäre Ziel soll Antworten auf noch zu spezifizierende topologisch-syntaktische Fragen auf Statistikenebene geben. Wie im weiteren Verlauf dargelegt wird, stellen u. a. die Filterungsoptionen zur Visualisierung und die statistischen Untersuchungen erhebliche Anforderungen an die Datenverwaltung. Diese Datenverwaltung wird ein Datenbanksystem (kurz DBS) übernehmen (müssen).

Methodisch besteht die Besonderheit der hier vorzunehmenden intrachromosomalen Untersuchungen im Fehlen von Vergleichssequenzen, d. h., die Untersuchungen basieren jeweils auf genau einem Chromosom. Auf jedem Chromosom sollen Sequenzbereiche mit hoher Übereinstimmung identifiziert werden, wobei besonders lange übereinstimmende Sequenzbereiche von biologischem Interesse sind.


Kapitel 2

Biologische Grundlagen und Formalisierungen

2.1 Biologische Grundlagen

2.1.1 Aufbau und Struktur der DNA

Die Zelle ist die kleinste autonom reproduzierte Grundeinheit aller lebenden Organismen; die niedersten Lebewesen (Einzeller) bestehen aus einer einzigen Zelle, alle höheren Lebewesen aus vielen (der Mensch z. B. aus rund 70 Billionen) Zellen (Vgl. [D91]).

Nach dem Mikrobiologen C. WOESE werden alle Lebewesen aufgrund ihrer Zelleigenschaften in drei Domänen unterteilt [W78]:

- Eubakterien
- Archaebakterien (Archaea)
- Eukaryonten

Als Eukaryo(n)t (eu - gut, echt; karyo(n) - Kern) werden alle Lebewesen (darunter auch der Mensch) mit einem speziellen Zellorganell, dem von einer Membran mit Poren umschlossenen Zellkern (Nukleus), zusammengefasst. Alle anderen Lebewesen werden als Prokaryo(n)ten (pro - vor) bezeichnet, da ihre Zellen keinen wohlausgeprägten Nukleus enthalten, allenfalls ein 'Nukleoid', ein kernartiges Gebilde ohne Membran. Es gibt jedoch andere Zellorganellen, wie bspw. Ribosomen, die sowohl in prokaryontischen als auch eukaryontischen Zellen vorhanden sind.

Im Nukleus des Menschen sind 23 Chromosomenpaare (diploider Chromosomensatz) enthalten, insgesamt also 46 Chromosomen. Betrachtet man die Chromosomen nicht paarweise, so bezeichnet man die 23 Chromosomen als einfachen (haploiden) Chromosomensatz.


In diesem Zusammenhang sei darauf hingewiesen, dass es nicht das (Human-)Genom gibt, extrachromosomale Erbinformationen befinden sich u. a. im Mitochondrion.
KAPITEL 2. BIOLOGISCHE GRUNDLAGEN UND FORMALISIERUNGEN

weil bspw. die Geschlechter nachweislich unterschiedliche Gonosomen haben. Würde ein Individuum das exakte Genom eines anderen Individuums besitzen, so würde man von einem Klon sprechen.

Es ist aber üblich, jeweils von dem Genom zweier verschiedener Spezies zu sprechen, sofern man im Kontext einer charakterisierenden Ebene bleibt.

Chromosomen und Plasmide bestehen - von ganz wenigen Ausnahmen abgesehen - aus kettentförmig angeordneten Desoxyribonukleinsäuren, kurz DNS. In Anlehnung an die englische Bezeichnung *deoxyribonucleic acid* verwendet man anstelle von DNS üblicherweise die Kurzform DNA.

Die DNA besteht aus Komponenten von drei verschiedenen chemischen Verbindungsklassen ([D99], [HH95]):

1. vier verschiedenen organischen Stickstoffbasen (kurz Basen),
2. einem Zucker (Desoxyribose), der eine chemische Verbindung mit einer Base eingehen,

1. Die Basen unterteilen sich in Pyrimidinbasen und Purinbasen. Pyrimidin ist ein Sechsringsystem mit zwei Stickstoffatomen und Purin ein Doppelringsystem, dass vier Stickstoffatome enthält. Die Pyrimidinbasen werden mit Cytosin (C) und Thymin (T) bezeichnet (s. Abb. 2.1 und 2.2) und die Purinbasen mit Guanin (G) und Adenin (A) (s. Abb. 2.3 und 2.4).

Abbildung 2.1: Cytosin

Abbildung 2.2: Thymin

Abbildung 2.3: Guanin

Abbildung 2.4: Adenin
D. h. einerseits, dass sich jeweils Adenin und Thymin bzw. Guanin und Cytosin paarweise setzt sich die DNA als über Wasserstoffbrücken verbundener Doppelstrang zweier zueinander. 1953 entwickelten J. D. WATSON und F. CRICK das Raummodell der DNA (s. Abb. 2.9 auf Seite 15). Danach zu Thymin bzw. Guanin zu Cytosin stets 1 : 1 und folglich konstant ist. 1951 stellte E. CHARGAFF bei allen DNA - Analysen fest, dass die Relation von Adenin ohne Sauerstoff, die Unterschiede sind in den Abb. 2.5 und 2.6 hervorgehoben. Wenn eine Base mit einer Desoxyribose über eine chemische Verbindung eine Einheit bildet, so wird diese Einheit als Nukleosid bezeichnet. Wird das Nukleosid durch Anlagerung eines Phosphorrestes erweitert, so wird diese chemische Verbindung als Nukleotid definiert (s. Abb. 2.8). (Desoxy-) Nukleotide sind die monomeren Bausteine der DNA, die ein lineares Polymer bilden. Eine Abfolge von Nukleotiden wird als DNA - Sequenz bezeichnet. 1951 stellte E. CHARGAFF bei allen DNA - Analysen fest, dass die Relation von Adenin zu Thymin bzw. Guanin zu Cytosin stets 1 : 1 und folglich konstant ist. 1953 entwickelten J. D. WATSON und F. CRICK das Raummodell der DNA (s. Abb. 2.9 auf Seite 15). Danach setzt sich die DNA als über Wasserstoffbrücken verbundener Doppelstrang zweier zueinander complementärer Nukleotidketten zusammen, deren Raumstruktur eine Doppelhelix bildet. D. h. einerseits, dass sich jeweils Adenin und Thymin bzw. Guanin und Cytosin paarweise
Abbildung 2.9: Die DNA-Doppelhelix mit komplementären Basenpaaren
durch Ausbildung spezifischer Wasserstoffbrücken gegenüber liegen (Basenpaar). Andererseits ist die Leserichtung des Komplementärstrangs entgegengesetzt. Zwar ist dieses Bauprinzip (fast) allen Lebewesen gemein, allerdings gibt es gravierende Unterschiede hinsichtlich Form (linear (humane Chromosomen), ringförmig) und Länge der Erbbträger. Die Genomlängen variieren zwischen mehreren 100000 Nukleotiden (Algen, Mycoplasmen), einigen Millionen Nukleotiden (Bakterien), einigen hundert Millionen Nukleotiden (Insekten), bis hin zu einer oder mehreren Milliarden(n) Nukleotiden (Vögel, Amphibien, Säugetiere) [L04].

I. A. steigt mit zunehmender Genomlänge auch die genetische Komplexität, auch wenn die Kröte *Xenopus* und der Mensch in etwa gleich lange Genome besitzen [L04].

2.1.2 Charakterisierung und biologische Interpretation der verwendeten Sequenzklassen

Bei den folgenden Begriffen wird darauf hingewiesen, dass diese in der einschlägigen Literatur durchaus abweichend definiert sein können. Hier sind die Literaturquellen [L04] und [S03] für die Begriffsdefinitionen ausschlaggebend.

Die Anordnung von vielen aufeinander folgenden Polynukleotiden (s. Def. 2.3 auf Seite 17) - im Folgenden als (DNA-) Subsequenz bezeichnet - einer chromosomalen DNA-Sequenz ist i. A. nicht zufällig. [L04] unterscheidet zwei Arten von DNA-Subsequenzen:

- **Nonrepetitive** DNA besteht aus unikalen Subsequenzen, d. h., es findet sich in einem haploiden Genom nur ein einzelnes Vorkommen dieser Subsequenz.

- **Repetitive** DNA enthält Subsequenzen, die in einem haploiden Genom mehrmals vorhanden sind:
  - **Moderat** repetitive DNA besteht aus relativ kurzen Subsequenzen, die üblicherweise 10 - 10000 mal im haploiden Genom wiederholt werden.

\[2\text{ im Sinne der Anzahl zusammenhängender Nukleotide (chromosomaler), oder insgesamt (kumuliert über alle Chromosomen)}\]

\[3\text{ Durch Genamplifikation entstandene Gensequenzen sind zwar per Definition repetitiv, aber genspezifisch und diesbezüglich im erweiterten Sinn ebenfalls unikal}\]
Hoch repetitive DNA setzt sich aus sehr kurzen (< 100 Bp) Subsequenzen zusammen, die 1000fach im haploiden Genom auffindbar sind.

Der Begriff *Satelliten* - DNA (kurz *Satelliten* oder auch *simple sequence dna*) wird heute oft als Synonym für hochrepetitive Sequenzen gebraucht [L04], [S03]. Satelliten

- finden sich in fast allen höheren eukaryontischen Genomen wieder
- bestehen aus sehr kurzen DNA- Sequenzen, die sich tandemartig mehrfach wiederholen
- können zytologisch in *heterochromatischen* Abschnitten nahe dem *Centromer* und den *Telomen* lokalisiert werden
- kann (bisher) keine bekannte Funktionalität zugeschrieben werden, sind aber von vielfältiger praktischer Bedeutung (z. B. bei forensischen Untersuchungen), da sie individuualspezifisch sind

Die heterogene Klasse der moderat repetitiven Sequenzen wird in

- SINE (short interspersed elements)
- LINE (long interspersed elements)

unterteilt. SINEs sind i. d. R. wenige 100 Bp lang, LINEs i. A. einige 1000 Bp. In vielen höheren Eukaryonten belegen sie ca. 50% der Genomsequenz. Im Humangenom machen 1,50 Mio. SINEs (0,85 Mio. LINEs) mit einer Länge von höchstens 0,3 kB (6 - 8 kB) einen Anteil von 15% (17%) aus [L04]. *Mikrosatelliten* und *Minisatelliten* bestehen aus noch kürzeren Repeatsequenzen als Satelliten: höchstens 10 Bp für Mikrosatelliten und einer dutzend- bis hundertfachen tandemartigen Wiederholung und ca. zehn bis fünfzig Bp für Minisatelliten mit zehn bis einhundert Wiederholungen. Diese beiden Satellitenarten sind genomweit verteilt und auch Bestandteil von RNA - Transkripten [4].

Die Genome von Prokaryonten bestehen (fast) ausschließlich aus nonrepetitiver DNA. Für niedere Eukaryonten besteht das Genom hauptsächlich aus nonrepetitiver DNA [L04]. Das Humangenom [5] besteht zu ca. einem Drittel aus repetitiver DNA [M03].

Repetitive DNA wird grobgranulär wie folgt klassifiziert:

- Repeats (maximal Repeats)
  - Tandems (Tandemrepeat)
  - supermaximal Repeat
- Palindrom (maximal Palindrom)

Die folgenden Definitionen von Repeats und Palindromen sind (hier) nur intrachromosomal sinnvoll: Z. B. lässt sich bei einer Translokation a posteriori die ursprüngliche Leserichtung des Translokats u. U. nicht / kaum mehr nachvollziehen, sodass die Ergebnisse einer interchromosomalen Repeat- oder Palindromberechnung wenig(er) aussagekräftig sind.

4 eine erstellte Kopie eines DNA - Abschnitts, bei der Thymin durch die Base Uracil ersetzt wird
5 gemeint ist das humane Referenzgenom von ENSEMBL
6 Verschieben eines DNA - Abschnitts von einem Chromosom zu einem anderen
2.2 Formalisierungen

Um die hier verwendeten vier repetitiven Sequenzklassen mit der gebotenen Genauigkeit zu definieren, bedarf es noch einiger formaler Definitionen.

**Definition 2.1** ($N_0, N^+$) Alle ganzen Zahlen im Bereich von $[0, \infty]$ ($[1, \infty[$) werden mit $N_0$ ($N^+$) gekennzeichnet.

**Definition 2.2** (Base ($\mathcal{B}, \mathcal{B}_N$)) Sei $\mathcal{B}$ bzw. $\mathcal{B}_N$ die Menge aller Basen die in einer DNA vorkommen. Dann ist $\mathcal{B}$ respektive $\mathcal{B}_N$ definiert als:

$$\mathcal{B} := \{A, C, G, T\},$$

$$\mathcal{B}_N := \{A, C, G, T, N\},$$

wobei $N$ als Platzhalter für $A$, $C$, $G$ oder $T$ fungiert.


**Definition 2.3** (Polynukleotid(-kette) ($\mathcal{B}^+, \mathcal{B}^1, \mathcal{B}_N, \mathcal{B}_N^+$)) Sei $\mathcal{B}^+$ aus Def. 2.2 gegeben. Dann bezeichne $\mathcal{B}^+$ ($\mathcal{B}^+_N$) die Menge aller (ggf. in Form von Nukleotiden) hintereinander verknüpften Basen aus $\mathcal{B}$ ($\mathcal{B}_N$), d. h.

$$\mathcal{B}^+ := B \cup \{AA, AC, AG, AT, CA, CC, CG, CT, GA, GC, GG, GT, TA, TC, TG, TT, \ldots\}$$

bzw.

$$\mathcal{B}^+_N := B_N \cup \{AA, AC, AG, AT, AN, CA, CC, CG, CT, CN, GA, GC, GG, GT, GN, TA, TC, TG, TT, TN, NA, NC, NG, NT, NN, \ldots\}$$

Sei $\mathcal{B}^1$ ($\mathcal{B}^1_N$) die echte Teilmenge von $\mathcal{B}^+$ ($\mathcal{B}^+_N$), $\mathcal{B}^1 \subset \mathcal{B}^+$ ($\mathcal{B}^1_N \subset \mathcal{B}^+_N$), die alle (ggf. als Nukleotide hintereinander verküpfte) Basen von $\mathcal{B}^+$ ($\mathcal{B}^+_N$) der Länge 1 enthält, d. h.

$$\mathcal{B}^1 = \mathcal{B}$$

$$\mathcal{B}^2 = \{AA, AC, AG, AT, CA, CC, CG, CT, GA, GC, GG, GT, TA, TC, TG, TT\}$$

$$\ldots = \ldots$$

7 s. Tabelle 6.1 auf Seite 22

17
Definition 2.4 (DNA - Sequenz \((S, S_N)\)) \(S_N\) heißt DNA - Sequenz gdw. \(S_N \in \mathcal{B}_N^+\). \(S\) heißt lückenlose DNA - Sequenz gdw. \(S \in \mathcal{B}_N^+\).

Definition 2.5 (Repeat, \(R, D_R, L_R\)) Seien \(S_1[i \ldots i+m] \in \mathcal{B}_m^{n+1}\) und \(S_2[j \ldots j+n] \in \mathcal{B}_n^{m+1}\) mit \(m, n \in \mathbb{N}_0\) und \(i, j \in \mathbb{N}^+\) und o. B. d. A. \(i < j\) DNA - Subsequenzen, die auf dem gleichen DNA - Strang liegen. \(S_1, S_2\) werden als Repeatarme des Repeats \(R\) bezeichnet, falls gilt:

\[
m + 1, n + 1 \geq R_{Lmin} \in \mathbb{N}^+ \quad (2.1)
\]
\[
m + 1 = l = n + 1 \Rightarrow S_1, S_2 \in \mathcal{B}^l \quad (2.2)
\]
\[
\forall k \in [0 \ldots l - 1] : S_1[i + k] = S_2[j + k] \quad (2.3)
\]
\[
S_1[i - 1] \neq S_2[j - 1] \land S_1[i + l] \neq S_2[j + l] \quad (2.4)
\]

Der Repeatarmabstand \(D_R\) ist definiert als \(j - (i + l)\) und die Repeatarmlänge \(L_R\) als \(l\). In Abhängigkeit von \(D_R\) und der Chromosomenlänge unterteilen sich die Repeats grobgranulär in folgende Kategorien:

\[
>> 0 \quad ('\text{random'} \text{ Repeat}) \quad (2.5)
\]
\[
> 0 \quad ('\text{normal'} \text{ Repeat}) \quad (2.6)
\]
\[
= 0 \quad (\text{Tandemrepeat}) \quad (2.7)
\]
\[
< 0 \quad (\text{overlapping Repeat}) \quad (2.8)
\]

Die Menge aller Repeats wird mit \(\mathcal{R}\) definiert.

(2.1) legt einen frei wählbaren Schwellwert \(R_{Lmin}\) für die Repeatarmlänge fest. Durch (2.2) gilt \(|S_1| = |S_2|\), mittels (2.3) sind beide Subsequenzen identisch und durch (2.4) sind sie maximal, d. h. nicht verlängerbar (s. Abb. 2.10 auf Seite 19).

Definition 2.6 (Tandem, \(\mathcal{T}\)) Ein Repeat \(R \in \mathcal{R}\), dass (2.1), (2.2), (2.3), (2.4) und (2.7) erfüllt, heißt Tandem. \(S_1, S_2\) werden dann als Tandemarme bezeichnet. Die Menge aller Tandems wird mit \(\mathcal{T}\) definiert.

Man beachte, dass wie in Abb. 2.10 auf Seite 19 dargestellt, die Subsequenzen auf dem Komplementärstrang in Bezug auf \(S_1, S_2\) ebenfalls Repeat- bzw. Tandemarme sind.

Als Vorgriff auf eine Verwendung eines supermaximal Repeats im folgenden Kapitel wird dieses hier definiert.

Definition 2.7 (supermaximal Repeat, \(S\)) Ein Repeat \(R \in \mathcal{R}\), dass (2.1), (2.2), (2.3), (2.4) erfüllt, sowie

\[
\neg \exists S_3[h, \ldots , h + k] \in B^{k+1}, k \in \mathbb{N}_0 : S_1 \leq S_3 \lor S_2 \leq S_3 \quad (2.9)
\]
erfüllt, heißt supermaximal Repeat. (2.9) besagt, dass es bezüglich derselben DNA-Sequenz keinen Repeatarm $S_3$ gibt, der $S_1$ oder $S_2$ als Subsequenz ($S_1 \leq S_3 \lor S_2 \leq S_3$) enthält. Die Menge aller supermaximalen Repeats wird mit $\mathcal{S}$ definiert.

Es sind zahlreiche biologische Mechanismen bekannt, die für die Existenz von Repeats bzw. Tandems verantwortlich sind. Willkürlich herausgegriffene Beispiele sind:

- Genduplikation / Genamplifikation (z. B. [S80], [BG02])
- unequal crossing-over ([L04]).

Während die Definition eines Repeats bzw. Tandems in der einschlägigen Literatur sehr homogen ist, variiert die (biologische) Definition eines Palindroms beträchtlich (vgl. [G99]), basiert aber auf der eindeutigen sprachwissenschaftlichen Definition: Ein Palindrom ist ein(e) ‘Wort oder Wortfolge das / die vorwärts wie rückwärts gelesen (den gleichen) Sinn ergibt’ (vgl. [D91]).

Die daraus abgeleitete und hier benötigte biologische Definition eines Palindroms lässt sich wohl als revers-komplementäres Repeat umschreiben.

**Definition 2.8 ((revers-komplementäres Repeat) Palindrom, $\mathcal{P}$, $D_P$, $L_P$)** Seien $S_1[i \ldots i + m] \in \mathcal{B}^{m+1}$ und $S_2[j \ldots j + n] \in \mathcal{B}^{n+1}$ mit $m, n \in \mathbb{N}_0, i, j \in \mathbb{N}^+$ und o. B. d. A. $i < j$ DNA-Subsequenzen, die auf dem gleichen DNA-Strang liegen. $S_1$, $S_2$ werden als Palindromarme des Palindroms $\mathcal{P}$ bezeichnet, falls gilt:

\[
\begin{align*}
  m + 1, n + 1 &\geq L_{\min} \in \mathbb{N}^+ & (2.10) \\
  m + 1 &= l = n + 1 \Rightarrow S_1, S_2 \in \mathcal{B}^l & (2.11) \\
  \forall k \in [0 \ldots l-1] : S_1[i + (l-1) + k] &= \text{compl}(S_2[j + k]) \quad (2.12) \\
  S_1[i - 1] &\neq \text{compl}(S_2[j - l]) \land S_1[i + l] &\neq \text{compl}(S_2[j - 1]) \quad (2.13)
\end{align*}
\]

Der Palindromarmabstand $D_P$ ist definiert als $j - (i + l)$ und die Palindromarmlänge $L_P$ als $l$. In Abhängigkeit von $D_P$ und der Chromosomenlänge unterteilen sich die Palindrome grobgranulär in folgende Kategorien:

\[
\begin{align*}
  >> 0 &\quad ('random' \text{ Palindrom}) \quad (2.14) \\
  \geq 0 &\quad ('normal' \text{ Palindrom}) \quad (2.15) \\
  < 0 &\quad (overlapping \text{ Palindrom}) \quad (2.16)
\end{align*}
\]

Die Menge aller Palindrome wird mit $\mathcal{P}$ definiert.

Abbildung 2.10: Beispiel für ein Repeat  
Abbildung 2.11: Beispiel für ein Palindrom
KAPITEL 2. BIOLOGISCHE GRUNDLAGEN UND FORMALISIERUNGEN

Durch Betrachtung des folgenden biologischen Zusammenhangs erklärt sich die Plausibilität von Def. 2.5 auf Seite 19: Restriktionendonukleasen bspw. erkennen charakteristische DNA-Subsequenzen mit der Folge einer anschließenden Durchtrennung der DNA an spezifischen Spaltstellen [L99]. Dadurch kann ein Stück doppelsträniger DNA herausgelöst und an anderer Stelle eingefügt werden (Translokation). Enthält dieses Stück einen Repeatarm, der durch 'falschen' Einbau sich nunmehr auf dem Komplementärstrang wiederfindet, wird aus dem Repeat ein Palindrom. In den Abb. 2.10 und 2.11 auf Seite 19 ist jeweils ein Ausschnitt von Abb. 2.9 auf Seite 15 mit einem Doppelstrang und komplementären Basenpaaren dargestellt. Wird z. B. in Abb. 2.10 der rechte Abschnitt des Doppelstrangs mit den Basen CGCA (TGC) auf dem oberen (unteren) DNA-Strang herausgetrennt und strangverkehrt eingefügt, so ergibt sich der in Abb. 2.11 abgebildete DNA-Doppelstrang.

Palindromische Subsequenzen sind biologisch auch bedeutsam:

- für die Sekundärstruktur ('Kleeblattstruktur') von tRNA’s (z. B. L04, SR95, H65)
- Aufspüren von Translokationen / Genome Rearrangements (z. B. P00)

Da Repeats und Palindrome i. w. S. Repeatstrukturen sind, nach den Def. 2.5 auf Seite 18 und 2.8 auf Seite 19 i. e. S. aber ein Repeat kein Palindrom ist, wird zu Gunsten einer scharfen semantischen / begrifflichen Trennung folgende Definition eingeführt:

**Definition 2.9 (Duplikat, D)** Unter einem Duplikat $D$ werden genau zwei Subsequenzen $S_1, S_2 \in B^l$ verstanden, die entweder nach Def. 2.5 auf Seite 18 Repeatcharakter oder nach Def. 2.8 auf Seite 19 Palindromcharakter besitzen. Die Menge aller Duplikate, $\mathcal{D}$, wird definiert als (Vgl. Abschnitt 6.2.2.2 auf Seite 71)

$$\mathcal{D} := \mathcal{P} \cup \mathcal{R} \cup \mathcal{S} \cup \mathcal{T}.$$

Nach der Festlegung der vier Duplikattypen bleibt noch die Frage zu klären, ob (effiziente) Suchalgorithmen zur Duplikatbestimmung existieren. Das Auffinden von zwei völlig identischen DNA-Subsequenzen $S_1 \in B^+ \leq S_N \in B^l_N$ und $S_2 \in B^+ \leq S_N \in B^l_N$ mit $S_1 = S_2$ erfolgt in linearer Zeit $\Theta(l)$, bspw. durch Anwendung des Boyer-Moore Algorithmus [BM77].

Aus biologischer Sicht sind (2.3) aus Def. 2.5 auf Seite 18 und (2.11) aus Def. 2.8 von Seite 19 sehr restriktiv. Z. B. können dadurch biologisch signifikante Duplikate aufgrund von mindestens einer zufälligen und irrelevanten Punktmutation verworfen werden. Ohne explizite Definition gehören nunmehr zwei DNA-Subsequenzen $S_1, S_2 \in B^l$ gleichfalls zu $\mathcal{R}(\mathcal{P})$, falls sie (2.3) (2.11)) 'fast' erfüllen. Folglich sind sie auch Elemente aus $\mathcal{D}$.

Damit scheiden Suchalgorithmen nach exakten Übereinstimmungen aus. Ein in Frage kommendes Suchverfahren ist die (optimale) lokale Sequenzalignierung.

**Definition 2.10 (optimale lokale Sequenzalignierung)** Seien $S \in B^l$ sowie $S_1 \in B^l = S[i \ldots i + m] \leq S_N \in B^l_N$ und $S_2 \in B^l = S[j \ldots j + m] \leq S_N \in B^l_N$ mit $i, j, m \in N^+, i \neq j$ gegeben. Sei $M_{i,j}$ die Punktzahl (Score) einer Bewertungsfunktion $f(S_1, S_2)$, die sich bei übereinstimmenden Basen $S_1[i+k] = S_2[j+k], 0 \leq k \leq m - 1$ erhöht und bei nicht übereinstimmenden Basen $S_1[i+k] \neq S_2[j+k], 0 \leq k \leq m - 1$ vermindert. Die Suche nach maximalen $M_{i,j}$ zu $i, j$ über eine vorgegebene Schwellenwertfunktion $f_T(T)(M_{i,j} \geq f_T(T))$ heißt optimale lokale Sequenzalignierung.

Für $T$ lässt sich u. a. eine Duplikaturmindestlänge $(R_{L_{\min}}, P_{L_{\min}})$ in $B_p$ einsetzen.
Kapitel 3
Berechnung der Duplikate

Als Datengrundlage für die Berechnung der Duplikate dienten alle 24 chromosomalen DNA-Sequenzen des humanen Referenzgenoms von ENSEMBL. Es bestand die Wahlmöglichkeit, die DNA-Sequenzen als Originalsequenz (unmasked) oder als vorverarbeitete Originalsequenz mit markierten low-complexity-Regionen (masked) zu beziehen. Aus Konsistenzgründen und einem vollständigen Datenabgleich zu anderen Lehrstuhlprojekten wurden sowohl die unmasked als auch die masked DNA-Sequenzen bezogen. Die 22 Autosomen und die beiden Gonorosen lagen im FASTA-Format vor.

Die Berechnung der Duplikate sollte durch eine optimale lokale Sequenzalignierung erfolgen. Dazu existiert ein Algorithmus, der von T. SMITH und M. WATERMAN entwickelt wurde, aber als Eingabe zwei verschiedene Sequenzen $S_1 \in B^+ \neq S_2 \in B^+$ erwartet. Für intrachromosomale Untersuchungen ist der Algorithmus nicht anwendbar, da die intrachromosomale Duplikatberechnung auf genau einer Sequenz erfolgt. Für mögliche interchromosomale Untersuchungen steht dem Vorteil der Optimalität aber ein Berechnungsaufwand von $\Theta(|S_1||S_2|)$ gegenüber. Für die vorliegenden Chromosomen und deren aus Tabelle 3.1 entnehmbaren Länge ist dieser Berechnungsaufwand jedoch zu groß.

Ein sehr verbreitetes Programm für die locale Alignierung ebenfalls paarweise verschiedener Sequenzen ist BLAST (basic local alignment search tool) [A90]. Es ist ein heuristisches Alignierungsprogramm, das möglicherweise 'nur' suboptimale Alignierungen findet und wurde insbesondere für die lokale Alignierung 'längerer' DNA-bzw. Proteinsequenzen konzipiert. Der Nachteil einer möglichen Suboptimalität wird in der Praxis für gewöhnlich durch ein deutlich besseres Laufzeitverhalten überkompensiert. BLAST fand bereits in der Studienarbeit [A03] Anwendung.

<table>
<thead>
<tr>
<th>Chromosom</th>
<th>DNA - Länge</th>
<th>Chromosom</th>
<th>DNA - Länge</th>
<th>Chromosom</th>
<th>DNA - Länge</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>246127941</td>
<td>9</td>
<td>136372045</td>
<td>17</td>
<td>81860266</td>
</tr>
<tr>
<td>2</td>
<td>243615958</td>
<td>10</td>
<td>135037215</td>
<td>18</td>
<td>76115139</td>
</tr>
<tr>
<td>3</td>
<td>199344050</td>
<td>11</td>
<td>134482954</td>
<td>19</td>
<td>63811651</td>
</tr>
<tr>
<td>4</td>
<td>191731959</td>
<td>12</td>
<td>132078379</td>
<td>20</td>
<td>63741868</td>
</tr>
<tr>
<td>5</td>
<td>181034922</td>
<td>13</td>
<td>113042980</td>
<td>21</td>
<td>46976097</td>
</tr>
<tr>
<td>6</td>
<td>170914576</td>
<td>14</td>
<td>105311216</td>
<td>22</td>
<td>49396972</td>
</tr>
<tr>
<td>7</td>
<td>158545518</td>
<td>15</td>
<td>100256656</td>
<td>X</td>
<td>153692391</td>
</tr>
<tr>
<td>8</td>
<td>146308819</td>
<td>16</td>
<td>90041932</td>
<td>Y</td>
<td>50286555</td>
</tr>
</tbody>
</table>

Tabelle 3.1: DNA-Längen der humanen Chromosomen in Bp (ENSEMBL V. 23)
KAPITEL 3. BERECHNUNG DER DUPLIKATE

3.1 BLAST

3.1.1 Kurze Beschreibung der Funktionsweise

Der BLAST-Algorithmus führt auf jeder Vergleichssequenz die folgenden drei Schritte aus (nach [W02]):

- Lokalisation der Hits:
  In der Vergleichssequenz werden Teilwörter der Länge \( w \) gesucht, die mit gleich langen Teilwörtern der Anfragesequenz eine Alignierung mit einem Wert größer \( T \) bilden. Eine derartige Alignierung wird *Hit* genannt.

- Expansion einer Hits:
  Ein Hit wird zu einer größeren lückenfreien Alignierung expandiert. Dazu wird die jeweils aktuelle Alignierung schrittweise nach links bzw. rechts um ein Zeichen erweitert. Die Erweiterung wird solange vorangetrieben, bis die entstehende Alignierung um einen festgelegten Wert \( X \) vom erweiterungslokalen Maximum abfällt. \( X \) wird als *drop-off* - Wert (dt. nachlassen, zurückgehen) bezeichnet. Dann stellt das lokale Maximum das Ergebnis dar und wird mit *HSP* (High-scoring Segment Pair) abgekürzt.

- Ausgabe der HSP’s:
  Hat ein HSP einen Wert größer als \( S \), wird er als lokale Alignierung ausgegeben. BLAST hat damit die Möglichkeit, mehrere lokale Alignierungen zu berechnen und auszugeben.

3.1.2 Vorverarbeitung und Anwendung

Für diese Diplomarbeit wurde NCBI-BLAST (*national center for biotechnology information*) in den Versionen 2.2.X unter Solaris (und AIX) verwendet. Die erforderliche Vorverarbeitung (Preprocessing) der Daten für das Programm *blastall* wurde für jedes Chromosom einzeln durch den Aufruf

```
formatdb -i <inputfile> -p F -o T
```
durchgeführt. Für die Beurteilung der Eignung von BLAST mit Blick auf eventuelle künftige Untersuchungen erfolgten Zeitmessungen unter Verwendung von */usr/bin/time*. Die Laufzeiten befinden sich im Anhang B.1 in der Tabelle B.1 auf Seite 124 und werden dort kurz analysiert und diskutiert.

Mit dem BLAST- Aufruf

```
blastall -p blastn -m 8 -g F F -i <inputfile> -d <databasefile> -o <outputfile>
```

erfolgte die Alignierung des durch die Vorverarbeitung erzeugten <databasefile> mit sich selbst (<inputfile>). Die 400 Ergebnisse für das vergleichsweise sehr kurze Y-Chromosom lagen nach 2,5 Stunden bei einer Prozessorauslastung von 99,7% vor. Der Hauptspeicherverbrauch wurde mit dem Unix-Kommando *top* mehrmals, in nicht-periodischen Abständen abgefragt: Er liegt bei ca. 1,6 GB und scheint weitestgehend konstant zu sein. Anzumerken ist, dass *blastall* (wie *bl2seq*) während der Berechnung offenbar aus Lauzeitgründen maximal 400 (AIX 800) High Scoring Segment Pairs (*HSP*‘s) (weiter-)verarbeitet, sodass als Ausgabe höchstens 400 (AIX 800) Endergebnisse vorliegen. Folgende Konsequenzen verbinden sich mit diesen Restriktionen:

1. zur Paramtererklärung s. BLAST - Dokumentation
2. s. man -S2 time

22
Die maximal 400 Duplikate dürften in Relation und je nach Chromosomenlänge (wenn überhaupt) nur wenige Promille der tatsächlich vorhandenen und signifikanten Duplikate ausmachen (vgl. Abschnitt 3.2).

Selbst diese maximal 400 Ergebnisse sind möglicherweise 'nur' suboptimal.

Duplikatreichtum einzelner Chromosomen ist weder erkennbar noch ggf. statistisch auswertbar (vgl. [SK03]).

Die Ausgabe trennt weder nach Repeats und Palindromen, noch nach Tandems.

Letztendlich scheitert die Duplikatbestimmung mit BLAST an der enormen Ressourcenbeanspruchung: Die Berechnung des X-Chromosoms wurde nach knapp 3 Stunden manuell abgebrochen, da die Prozessorauslastung zwar rechnerisch bei knapp 10%, die aperiodisch beobachtete effektive Prozessorauslastung aber bei weit unter 1% lag, deren Ursache sehr wahrscheinlich ein (ebenfalls kurzfristig beobachteter) Hauptspeicherverbrauch von ca. 8,3 GB war.

Eine vollständige Duplikatberechnung bei zumindest quantitativ unbefriedigenden Ergebnissen hätte schätzungsweise mehrere Monate in Anspruch genommen, die im Rahmen dieser Diplomarbeit nicht zur Verfügung standen. BLAST ist hier zur intrachromosomalen Duplikatbestimmung nicht geeignet.

3.2 VMATCH

3.2.1 Kurze Beschreibung der Funktionsweise

VMATCH basiert auf Enhanced Suffix Arrays, die in [AKO02] und [AOK02] detailliert beschrieben werden. Ein Enhanced Suffix Array ist eine Erweiterung des Suffix Arrays, das in Anlehnung an [G99] nachfolgend definiert wird:

**Definition 3.1 (Suffix Array)** Sei \( T[m] \) eine Zeichenkette, bspw. \( T[m] \in \mathbb{B}^m \). Ein Suffix Array für \( T \), Pos genannt, ist ein Array der ganzen Zahlen im Bereich von 1 bis m, die die lexikographische Reihenfolge von \( m \) Suffixe von \( T \) angeben.

D. h., der Suffix von \( T \), der an Position \( Pos(1) \) beginnt, ist der lexikographisch kleinste Suffix. I. A. gilt ferner, dass der Suffix \( Pos(i) \) von \( T \) kleiner ist als der Suffix \( Pos(i + 1) \). Ein Enhanced Suffix Array ist ein Suffix Array mit einer \( lcp \)-Tabelle [AKO02]. Der entscheidende Vorteil eines Enhanced Suffix Arrays liegt im deutlich geringeren Speicherverbrauch (s. u.). Dadurch lassen sich einerseits z. B. erst sehr lange Chromosomen (mehrere 100 Mbp) verarbeiten, andererseits verbessert sich die Laufzeit teilweise erheblich.

3.2.2 Vorverarbeitung und Anwendung


- deutlich geringerer Hauptspeicherverbrauch und (dadurch) beschleunigte Ausführungszeiten (zumindest in Bezug auf die Duplikatberechnungen, nicht unbedingt auf die notwendige Vorverarbeitung)

---

3s. auch Definition 6.2 auf Seite 72
KAPITEL 3. BERECHNUNG DER DUPLIKATE

- keine quantitative Beschränkung von Ergebnissen
- Gewissheit der Optimalität und Vollständigkeit der Ergebnisse
- Trennung der Berechnung nach Sequenzklassen
- obligatorische Angabe der Mindestlänge von Duplikatarmen


Unter der Voraussetzung eines ressourcenschonenderen Hauptspeicherverbrauchs umfasste die Berechnung aller Duplikate für jedes Chromosom also

- Repeats
  - Tandems
  - supermaximal Repeats

- Palindrome

Wegen der obligatorischen Angabe einer Mindestlänge für jeden Duplikatarm bedurfte es zunächst der Festlegung geeigneter Größen(-ordnungen). Mit der Wahl von restriktiven (großzügigen) Parametern ist einerseits der Vorteil der Sensitivität (Selektivität) verbunden, andererseits besteht der Nachteil einer geringen Selektivität (Sensitivität).

Eine ausführliche Diskussion über die Entscheidungsfindung zur Bestimmung der verwendeten VMATCH - Parameter befindet sich in Abschnitt 3.3 auf Seite 27. Eine vorläufige Einschätzung zu den Ergebnissen aller Duplikatarten enthält Abschnitt 3.5 auf Seite 30. Für die Berechnung von Tandems und supermaximalen Repeats ist nur die Angabe der Mindestlänge (-l <value>) zulässig. Eine Möglichkeit, in begrenztem Umfang Mismatches oder Gaps (dt. Lücke) zu erlauben, wird nicht unterstützt. D. h., dass die Tandemarme und supermaximalen Repeatarme jeweils vollkommen identisch sind. Die entsprechenden VMATCH - Aufrufe lauten:

- Tandems:
  \[ \text{vmatch -tandem -l 75 <chromosomenname>} \]  
  (= PID 6)

- supermaximal Repeats:
  \[ \text{vmatch -supermax -l 100 <chromosomenname>} \]  
  (= PID 5)

Bei der Berechnung von Repeats und Palindromen entfällt die Beschränkung der exakten Übereinstimmung der Duplikatarme. Durch Angabe optionaler Parameter kann die Repeat-

4Bei Benutzung von \text{bl2seq} ist keine Vorverarbeitung nötig
5d. h., dass bei der Berechnung Basen eingefügt werden dürfen, wenn dies für die Alignierung vorteilhaft ist. Die Kennzeichnung einer solch eingefügten Base ist dem Ausfüllen einer Lücke vergleichbar.
6entspricht \text{param}_{id} - Attributwerten in den Tabellen 4.6 auf Seite 37 bis 4.9 auf Seite 39.
KAPITEL 3. BERECHNUNG DER DUPLIKATE

bzw. Palindromeberechnung jeweils auf (ausschließlich) Mismatches bzw. Mismatches und Gaps ausgeweitet werden. Im ersten (zweiten) Fall wird in Anlehnung an die \textit{Hammingdistanz} (\textit{Editdistanz}) der Parameter \texttt{-h <distance_value>} (\texttt{-e <distance_value>}) verwendet. In beiden Fällen ist die Angabe einer \texttt{seedlength} mittels \texttt{-seedlength <seed_value>} zwingend. Nähere Informationen, insbesondere zur Korrelation zwischen \texttt{length}, \texttt{distance_value} und \texttt{seed_length}, stehen im Handbuch zu VMATCH \cite{K03}.

Zur Beantwortung der Frage, ob und ggf. in welchem Umfang die Parameterwahl die Repeat- bzw. Palindromergebnisse quantitativ und qualitativ (im biologisch funktionellen Kontext) beeinflusst, wurden bei der Hammingdistanz und der Mindestlänge tendenziell extremale Werte benutzt. Sollte sich wider Erwarten herausstellen, dass die Parameterwahl zu sensitiv ist, so liessen sich die Daten in der Ausgabedatei noch nachträglich filtern. Eine eher (zu) selektiv angelegte Parameterwahl lässt sich nur durch eine vergleichsweise aufwendigere, erneute VMATCH - Berechnung korrigieren.

Die Editdistanz blieb aufgrund partieller (von Gaps verursachter?) heterogener Angaben zu den Duplikatarmängen unberücksichtigt. Die Berechnungen erfolgten durch die vier VMATCH - Aufrufe:

- **Repeats:**
  
  \begin{align*}
  \text{vmatch} & \text{-d} \text{-l 100} \text{-h 1} \text{-seedlength 50} <\text{chromosomenname}> \quad (= \text{PID 1}) \\
  \text{vmatch} & \text{-d} \text{-l 200} \text{-h 10} \text{-seedlength 40} <\text{chromosomenname}> \quad (= \text{PID 2})
  \end{align*}

- **Palindrome:**
  
  \begin{align*}
  \text{vmatch} & \text{-p} \text{-l 100} \text{-h 1} \text{-seedlength 50} <\text{chromosomenname}> \quad (= \text{PID 3}) \\
  \text{vmatch} & \text{-p} \text{-l 200} \text{-h 10} \text{-seedlength 40} <\text{chromosomenname}> \quad (= \text{PID 4})
  \end{align*}

Für diese vier VMATCH - Aufrufe sind die Ergebnisse in Abb. 3.1 auf Seite 26 veranschaulicht. Zunächst fällt auf, dass für jedes Chromosom die Anzahl der Palindrome und Repeats (bei sonstiger gleicher Parameterwahl) in etwa gleich ist, großteils mit einer Abweichung im einstelligen Promillebereich.

Die Beantwortung der Frage, ob der chromosomale Gleichstand dieser Duplikatanzahl humangpezifisch ist, durch biologische Mechanismen verursacht wird bzw. auf eine andere Erklärung zurückzuführen ist, zieht die in Abschnitt 3.4 auf Seite 28 besprochenen Nachuntersuchungen mit sich.

### 3.2.3 Ausgabetransformation über C - Schnittstelle

VMATCH bietet dem Anwender die Möglichkeit, die Ergebnisse zu modifizieren oder die Ausgabe in beschränktem Maße zu transformieren, z. B. um die Ergebnisse an ein bestehendes Schema einer DB anzupassen. U. U. reichen diese Bordmittel jedoch nicht aus, sodass ein externe Nachbearbeitung (Postprocessing) erfolgen muss. Dies kann auf zwei Wegen geschehen:

Da die Ausgabe der durch VMATCH berechneten Ergebnisse standardmäßig auf dem Bildschirm (\textit{stdout}) erfolgt, lassen sich die Daten einerseits über eine Pipe \footnote{Unter einer Pipe (dt. Röhre) versteht man die Verkettung von hintereinander ausgeführten Befehlen, bei denen jeweils die Ausgabedaten des einen Befehls auch die Eingabedaten des nachfolgenden Befehls sind.} und bspw. durch Unix - / Solaris - Programme wie \textit{sed} oder \textit{awk} transformieren und in eine Datei umleiten. 

25
Andererseits kann man die Ergebnisse erst in eine Datei umleiten und anschließend mit externen Programmen bearbeiten. Unter Effizienzaspekten ist erstere Variante zu bevorzugen, da sie weniger I/O intensiv ist.

VMATCH offeriert mittels der *selection functions* die Möglichkeit, über eine C-Schnittstelle eine Bibliothek (Unix: *shared object*; Windows: *library*) zu programmieren. Die Bibliothek kann auch Parameter entgegennehmen und wird quasi als VMATCH-Parameter angegeben, sodass bspw. über den Aufruf

\texttt{vmatch \textasciitilde tandem \textasciitilde l 75 -selfun vmatch2db2.so <p1> <p2> <chromosom>}

VMATCH über den Parameter *-selfun* mitgeteilt wird, dass nachfolgender Parameter der Name (*vmatch2db2.so*) einer Bibliothek ist und dieses in diesem Fall wiederum zwei Parameter *<p1>* und *<p2>* entgegen nimmt. Unter Berücksichtigung des noch vorzustellenden Datenbankschemas wird *<p1>* mit einem frei wählbaren Identifier (hier: 1 für Chr. 1, 2 für Chr. 2, \ldots, 23 für Chr. X, 24 für Chr. Y) belegt. *<p2>* wird durch einen weiteren Identifier ersetzt, der alle verwendeten VMATCH-Parameter umfasst (bspw. 1 für *-tandem -l 75*, 2 für *-supermax -l 100*, usw.). Dadurch kann jedes VMATCH-Ergebnis (genau) einem Chromosom und (genau) einem VMATCH-Lauf eindeutig zugeordnet werden.

Während der Entwicklungsphase musste ein Solaris-Skript die gewünschte Transformation
in das benötigte Datenbankschema übernehmen. Anschließend wurde dies komfortabel und zeitsparend 'on - the - fly' von obiger Bibliothek erledigt.

3.3 Parameterdiskussion

Die Abwägung unterstützte ein statistischer Ansatz, mit dem Ziel der Bestimmung einer unteren Schranke. Die zugrunde liegende Idee soll am folgenden Beispiel illustriert werden:

Sei $\mathcal{S} \in \mathbb{B}^5$ gegeben. $\mathcal{S}$ enthält mindestens ein $\mathcal{R} \in \mathbb{R}$ mit Repeaterlänge $L_R = 1$, weil bei vier verschiedenen Basen (A, C, G, T) die fünfte dann eine von diesen vier Basen sein muss. In diesem Fall ist $\mathcal{R}$ mit $R_L = 1$ biologisch irrelevant, weil es statistisch 'erzwungen' wird. D. h., bei $\mathcal{S}$ stellt eine Repeaterlänge $L_R \geq 2$ eine untere Schranke für Signifikanzaspekte dar.

Für die Bestimmung der unteren Schranke im allgemein gültigen Fall sei eine DNA-Sequenz $\mathcal{S}[1..n] \in \mathbb{B}^n$ gegeben. Offensichtlich besitzt $\mathcal{S}$ zwei Subsequenzen $\mathcal{S}_1[1..n - 1] \in \mathbb{B}^{n-1}$ und $\mathcal{S}_2[2..n] \in \mathbb{B}^{n-1}$, drei Subsequenzen $\mathcal{S}_I[1..n - 2] \in \mathbb{B}^{n-2}$, $\mathcal{S}_I'[2..n - 1] \in \mathbb{B}^{n-2}$ und $\mathcal{S}_I''[3..n] \in \mathbb{B}^{n-2}$, ..., $n$ Subsequenzen $\mathcal{S}_k[k..k] \in \mathbb{B}^1$ mit $k, n \in \mathbb{N}^+, 1 \leq k \leq n$.

Für $n = 5$ ist durch obiges Beispiel bereits bekannt, dass es immer ein Repeat mit Repeaterlänge $R_L = 1$ gibt. Ist $n = 18$ (67), so gibt es 17 (65) Repeats mit $R_L = 2$ (3). Da es aber $|\mathbb{B}^{R_L=2}| = 16 < 17$ ($|\mathbb{B}^{R_L=3}| = 64 < 65$) verschiedene Repeats mit $R_L = 2$ (3) gibt, muss mindestens ein Repeat mit $R_L = 2$ (3) ex. und folglich gilt für die untere Schranke $S_u^n \in \mathbb{N}^+$:

$$S_u^{18} = 3 (S_u^{67} = 4).$$

Für beliebiges, aber festes $n$ mit $S_u^n \ll n$ gilt allgemein:

$$|\mathbb{B}^1|S_u^n + (S_u^n - 1) = n \quad 4^{S_u^n} = n + 1 - S_u^n \quad 4^{S_u^n} \approx n \quad S_u^n = \log_4 n$$

Setzt man für $n$ die Länge des kürzesten (längsten) Chromosoms (21) (21) ein, $n = 46976097$ (246047941), so ergibt sich als untere Schranke eine Mindestlänge von 13 (14). Als Richtschnur für die Festlegung der Mindestlänge waren aber folgende Sachverhalte von entscheidender Bedeutung (s. Abschnitt 2.1.2 auf Seite 15):

1. die Länge und Anzahl von SINES (und LINEs) im Humangenom
2. die Gesamtlänge von Mikrosatelliten und Minisatelliten sowie jeweils die Länge ihrer Repeatsequenzen
3. die durchschnittliche Länge und Anzahl von Exons im Humangenom
4. die Länge und Sekundärstruktur von tRNA - Sequenzen

die nachfolgend erläutert werden:

1. Die 1,5 Mio. SINES mit einer Maximallänge von 300 Bp lassen auf sensitive Ergebnisse bei Wahl einer Duplikatarmmindestlänge im 100er Längenbereich schließen.
2. Bei Wahl einer Duplikatarmmindestlänge im hohen zweistelligen oder niedrigen dreistelligen Zahlenbereich sollten auch Mikro- und Minisatelliten gefunden werden, ein Herunterbrechen auf die Länge (max. 10 Bp) der Repeateinheiten von Mikrosatelliten kommt aus den obigen statistischen Erwägungen nicht in Frage, dies gilt prinzipiell auch für die Repeateinheiten (Länge 10 - 50 Bp) der Minisatelliten.


4. Die Sekundärstruktur - in Form eines Kleeblatts mit drei 'Blättern' und einem Stil - einer tRNA - Sequenz wird durch deren palindromische Eigenschaft hervorgerufen und setzt sich aus 74 - 95 Basen zusammen. Insbesondere wegen eines fünften 'Blatts' (extra arm) mit einer nicht - palindromischen Sequenz und variable Länge von drei bis einundzwanzig Basen, dürfte das Setzen der Mindestlänge eines Palindromarmes auf gut 70 Basen zum Auffinden von tRNA - Sequenzen erfolglos bleiben.

Diese Gründe führten zur Festlegung einer Duplikatmindestlänge (außer Tandems) von 100 Basen bei max. einem Mismatch und seedlength 40 bzw. 200 Basen bei max. zehn Mismatches und seedlength 50. Folglich sind Duplikatarmidentitäten von mindestens 99% bzw. mindestens 95% zu erwarten. Aufgrund der nach Def. 2.6 auf Seite 18 erzwungenen lückenlosen Anordnung von Tandemarmen wird in diesem Fall die Mindestlänge der Tandemarme auf 75 Basen heruntergesetzt. Die seedlength orientierte sich am oberen Ende der Länge der Repeateinheiten von Minisatelliten.

3.4 Nachuntersuchungen zur Repeat- und Palindromanzahl

Die Nachuntersuchungen zur Aufdeckung der Ursache der chromosomenweise (fast) identischen Anzahl von Repeats und Palindromen erstrecken sich auf die Beantwortung der Fragen

Ist der 'Gleichstand' der Anzahl

1. . . . humanspezifisch?
2. . . . statistisch bedingt?
3. . . . biologisch verursacht?


Abgesehen von Besonderheiten wie der singulären oder deutlichen Abweichung der Anzahl Repeats und Palindrome bei Chromosom 10 von *mus musculus* und einer vergleichsweise exorbitanten Anzahl 3 von Repeats und Palindromen für das X - Chromosom dieser Spezies, Wegen einer maximal erlaubten und erreichten Dateigröße von 2 GB enthalten diese beiden Ergebnisdateien mit Sicherheit nicht die vollständigen Ergebnisse
KAPITEL 3. BERECHNUNG DER DUPLIKATE

Abbildung 3.2: Anzahl berechneter Duplikate (ohne Tandems und supermax. Repeats)

spiegelt sich der Gleichstand wider. Er ist daher nicht humanspezifisch.

(2) Eine andere denkbare Erklärung für den Gleichstand ist die VMATCH - Parameterwahl (zu sensitiv, zu unselektiv). Eine generelle, restriktivere VMATCH - Parameterwahl steht wie am Kapitelanfang geschildert nicht zur Debatte. Andernfalls besteht die Gefahr, möglicherweise schon errechnete intragenische / intragenische Duplikate herauszufiltern und hauptsächlich intergenische Duplikate (Transposons? ([L04])) bestimmt zu haben.

(3) Wie bereits im vorangegangenen Kapitel aufgeführt, gibt es biologische Mechanismen (Genduplikation, unequal crossing - over, ...), die mitverantwortlich für die Anzahl der Repeats und Palindrome sein können. Allerdings übersteigt die Anzahl der Duplikate die (bisher) bekannten Gene und Exons bei weitem, sodass man wohl mehrere Ursachen in Erwägung ziehen muss.

Alternativ finden daher maskierte DNA - Sequenzen Verwendung, mit dem Nachteil vielfältiger Maskierungsmöglichkeiten. Weil ENSEMBL pro Chromosom genau eine maskierte DNA - Sequenz zur Verfügung stellt, erübrigt sich eine Diskussion zur 'richtigen' Auswahl der verwendeten, maskierten DNA - Sequenz. Die Ergebnisse sind für die Repeats und Palindrome bei gleichen VMATCH - Parametern in Tabelle C.7 auf Seite 134 angegeben und in Abbildung 3.2 visualisiert.
Es ist ersichtlich, dass im Vergleich zu den unmaskierten DNA-Sequnzen jeweils nur ein Bruchteil an Repeat- und Palindromergebnissen anfällt und unabhängig von der Achskalierung bei Vergleich mit Abbildung 3.1 auf Seite 26 (teilweise skalierungsbedingt) kein Repeat- und Palindromgleichstand zu erkennen ist.

3.5 Kapitelzusammenfassung

Alle Duplikate wurden durch intrachromosomale Berechnungen bestimmt. Für intrachromosomale Berechnungen dieser Größenordnung hat sich VMATCH bezüglich des deutlich geringeren Hauptspeicherverbrauchs, der Laufzeit und der Quantität der Ergebnisse (zumindest gegenüber BLAST) als Mittel der Wahl erwiesen.

Darüber hinaus bietet VMATCH aber auch interchromosomale Betrachtungen an: maximum substring matches, maximum unique matches (mum) und complete matches (s. [K03]). Bei 24 Chromosomen und $\frac{24(24-1)}{2} = 276$ paarweise verschiedenen interchromosomalen Vergleichen, sind solche Untersuchungen aufgrund der Anzahl der Ergebnisse wohl bestenfalls für Tandems vollständig praktizierbar. Ansonsten sind entweder geeignete Datenreduktionen vorzunehmen oder nur ausgewählte Betrachtungen.

Die Tandems und supermaximal Repeats stellen nur einen Bruchteil der Duplikate dar; fast alle Duplikate sind entweder Repeats oder Palindrome. Bemerkenswert ist aber die Relation der Ergebnisse von Repeats und Palindromen, insbesondere bei den unmaskierten DNA-Sequnzen: Diese sind mit insgesamt 16619709 Repeats zu 16489529 Palindromen und 45462559 Repeats zu 45270907 Palindromen nicht nur insgesamt fast identisch, sondern im Wesentlichen auch chromosomenweise.

Ferner ist festzustellen, dass i. A. bei allen Duplikatarten die Anzahl der Ergebnisse mit der Länge der Autosomen korreliert. Die Gonosomen scheinen diesbezüglich eine Ausnahmestellung zu beanspruchen.
Kapitel 4

Logischer Entwurf der DB SYNTENY

Den detaillierten Erläuterungen des Datenbankschemas der DB SYNTENY (s. u.) in Verbindung relevanter Datenbankobjekte wie Trigger (dt. Auslöser) in den einzelnen - gleich kurz vorgestellten - Abschnitten geht vorher noch eine kurze Diskussion über das Für und Wider des Einsatzes eines (relationalen) DBS voraus.


Ein Anliegen dieser Diplomarbeit ist die Untersuchung der Duplikate auf topologische und funktionelle Eigenschaften (s. Kapitel [7] auf Seite [83]). Da weder die Duplikatergebnisse noch die verwendeten DNA-Sequenzen im FASTA-Format topologische respektive funktionelle Daten beinhalten, müssten diese noch bezogen werden. Diese und weitere Informationen bietet ENSEMBL in diversen Dateien an, deren Datenstruktur auf einem relationalen Datenbankschema beruht. ENSEMBL nennt die betreffende DB HS_CORE.

Eine Verwaltung der Duplikatergebnisse und HS_CORE-Daten in zwei separaten Datenbanken - SYNTENY und HS_CORE - drängt sich im Hinblick auf andere, bereits am Lehrstuhl vorhandene DB biologischen Inhalts, nahezu auf. Daraus ergeben sich u. a. auch einige Vorteile:

- einheitlicher Datenzugriff / Datenmanipulation über SQL
- deutlich reduzierter Entwicklungs- / Programmieraufwand
- vereinfachte Erweiterung durch Hinzufügen weiterer DB (von anderen Spezies, . . .)
- bereits existierende DB lassen sich ggf. für Analysezwecke mit modifizierten Zielsetzungen ausnutzen
- Performanzsteigerung durch Verwendung von Indizes

Nicht zu unterschätzende Nachteile sind insbesondere:

1Der Datenumfang bereitete neben der Ressourcenbeschränkung und der fehlenden Erfahrung im Umgang mit solchen Datenmassen sowie einigen Unzulänglichkeiten des DBMS (nach Auffassung des Autors) ernsthafte Probleme.
• Entwurf eines eigenen Datenbankschemas unter Berücksichtigung von Erweiterungsaspekten

• notwendige Speicherplatzkalkulationen für Tabellenbereiche wegen des Datenumfangs

• detaillierte(re) Kenntnisse in der Administration und Konfiguration von DBS

• erhöhter Ressourcenbedarf (Hauptspeicher: Pufferpools / Sekundärspeicher: Tabellenbereiche (s. Kapitel 5 auf Seite 48)

SYNTENY (dt. Syntänie) ist eine von den drei in Abb. 4.1 angegebenen DB. Kapitel 7 auf Seite 83 widmet sich den DB HS_CORE und FED. Obwohl sie fast ausschließlich VMATCH - Ergebnisse beinhaltet, wurde für diese DB der Name SYNTENY verwendet, um eine potentielle Verwechslungsgefahr mit dem Programm VMATCH auszuschließen.

Syntänie ist ein biologischer Begriff und wird in Definition 4.1 festgelegt:

Definition 4.1 (Syntänie) Syntänie kennzeichnet die Konservierung von mehreren aufeinanderfolgenden Genen zwischen zwei Spezies. (Vgl. [PHF99]).

Zwar besteht das Ziel dieser Diplomarbeit nicht primär in der Aufdeckung von Syntäniern. Dennoch ist diese Namensgebung nicht unangebracht, weil sie darauf ausgedehnt werden kann.

4.1 Schemata und Schemanamen

Schemata sind von DB2 verwendete Datenbankobjekte, um andere Datenbankobjekte logisch zu gruppieren. Die meisten Datenbankobjekte werden nach einer zweiteiligen Namenskonvention schema_name.object_name benannt [BW03]. Wie man im weiteren Verlauf sehen wird, lassen sich viele Datenbankobjekte (insbesondere von SYNTENY) bis auf wenige Ausnahmen logisch gruppieren.

Mit der Verwendung von Schemanamen verbindet sich bspw. eine bessere semantische Identifizierung von Objektnamen, deren Länge auf teilweise maximal 18 Zeichen begrenzt ist, was hier aufgrund vieler erstellter Prozeduren und Funktionen oft zu einer ’kryptischen’ Namensgebung führt. Auch Utilities wie REORGCHK lassen als Parameter Schemanamen
Abbildung 4.2: Direkte logische Abhängigkeiten zwischen den Schemata

zu, sodass anstelle von (einzeln) Datenbanktabellen (kurz DBT) gleich alle DBT im zugehörigen Schema bearbeitet werden können.

Für zu erstellende Datenbankobjekte (DBT, Indizes, Views, Trigger, ...) von SYNTENY werden die Schemanamen

- **duplicates**: logische Gruppierung der VMATCH - Ergebnisse
  (s. Abschnitt 4.2.3 auf Seite 43)

- **admin**: logische Gruppierung allgemeiner Spezieserträge und Versionen
  (s. Abschnitt 4.2.1 auf Seite 34)

- **sequence**: logische Gruppierung von sequenzspezifischen Daten
  (s. Abschnitt 4.2.2 auf Seite 40)

- **results**: logische Gruppierung von abgespeicherten Resultaten
  (s. Abschnitt 4.2.5 auf Seite 46)

verwendet. Abb. 4.2 verdeutlicht die direkten (logischen) schemataübergreifenden Zugriffe auf bestimmte DBT, die u. a. zur Wahrung der Datenintegrität erforderlich sind.

### 4.2 Datenbanktabellen

Die DB **SYNTENY** enthält im Wesentlichen die

- rund 125 Mio. Duplikate des Referenzgenoms des *homo sapiens*
- rund 40 Mio. Duplikate des Referenzgenoms des *pan troglodytes*

Abbildung 4.3: Datenbankschema der Administrationskomponente


4.2.1 Administrationskomponente


4.2.1.1 `admin.species`

Die DBT `admin.species` verwaltet alle Speziesinträge und besteht aus den folgenden drei Attributen

- `species_id` (smallint)
KAPITEL 4. LOGISCHER ENTWURF DER DB SYNTENY

- \textit{species\_name} (varchar(40))
- \textit{species\_alt\_name} (varchar(25))

Jeder Species, die in \textit{SYNTENY} verwaltet wird oder verwaltet werden soll, erhält unter \textit{species\_name (species\_alt\_name)} ihren wissenschaftlichen Namen (Trivialnamen) zugewiesen. Mittels des Primärschlüssels \textit{species\_id} wird jede Spezies mit einer eindeutigen Nummer belegt. Beide Namen müssen aufgrund einer \textit{unique} - Restriktion syntaktisch eindeutig sein. Tabelle 4.1 gibt zur Veranschaulichung einige Datensätze der DBT \textit{admin.species} wieder. Ein Trigger für die Lückenlosigkeit von \textit{species\_id} und die Kleinschreibung von \textit{species\_name} und \textit{species\_alt\_name} wurde erstellt.

\begin{center}
\begin{tabular}{|c|c|c|}
\hline
\textit{species\_id} & \textit{species\_name} & \textit{species\_alt\_name} \\
\hline
1 & homo sapiens & human \\
2 & mus musculus & mouse \\
3 & pan troglodytes & chimpanzee \\
\hline
\end{tabular}
\end{center}

Tabelle 4.1: Die Datensätze der DBT \textit{admin.species}

4.2.1.2 \textit{admin.version}

Wie weiter oben erwähnt, erfolgte die Entwicklung anhand der ENSEMBL - Version 18\_34 und wird im Zeitraum des Aufschreibs dieser Diplomarbeit an der aktuellen ENSEMBL - Version 23\_34 nachvollzogen. Um diesem Umstand Rechnung zu tragen und pro Spezies auch mehrere Versionen verwaltbar zu machen, wurde die DBT \textit{admin.version} mit den beiden Attributen

- \textit{version\_id} (smallint)
- \textit{version\_description} (varchar(40))


Ein Trigger für die Lückenlosigkeit von \textit{version\_id} und die Kleinschreibung von \textit{version\_description} wurde kreiert.

\begin{center}
\begin{tabular}{|c|c|}
\hline
\textit{version\_id} & \textit{version\_description} \\
\hline
1 & ensembl\_version\_18\_34 \\
2 & ensembl\_version\_23\_34 \\
3 & ensembl\_version\_23\_33 \\
4 & ensembl\_version\_23\_1 \\
\hline
\end{tabular}
\end{center}

Tabelle 4.2: Auszug aus der DBT \textit{admin.version}

\footnote{Duplikatdaten zu mus musculus sind wie eingangs erwähnt nicht enthalten, die Aufführung dient lediglich zu Verständniszwecken}
4.2.1.3 admin.species_version

Die DBT *admin.species_version* setzt sich aus den beiden Attributen

- *species_id* (smallint)
- *version_id* (smallint)


<table>
<thead>
<tr>
<th>species_id</th>
<th>version_id</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
</tr>
</tbody>
</table>

Tabelle 4.3: Auszug aus der DBT *admin.species_version*

Bezugsnehmend auf die beiden Tabellen 4.1 auf Seite 35 und 4.2 auf Seite 35 bedeuten die vier Einträge in Tabelle 4.3, dass die Spezies *homo sapiens* in den ENSEMBL-Versionen 18.34 und 23.34 vorliegt. Dagegen liegt die Spezies *pan troglodytes* ‘nur’ in der ENSEMBL-Version 23.1 vor.

4.2.1.4 admin.vmatch_seqtypes

Diese DBT nimmt über das Attribut

- *sequence_type* (varchar(20))

...die syntaktisch eindeutigen Namen der in VMATCH verwendbaren Sequenzklassen auf, wobei wiederum ein Trigger die Kleinschreibung von *sequence_type* erzwingt. Da die zulässigen Attributwerte schon während der Erstellung der DBT vorgegeben sind, dient der Trigger zur Fehlervermeidung durch Attributwerte mit Großbuchstaben.

<table>
<thead>
<tr>
<th>sequence_type</th>
</tr>
</thead>
<tbody>
<tr>
<td>repeat</td>
</tr>
<tr>
<td>tandem</td>
</tr>
<tr>
<td>palindrome</td>
</tr>
<tr>
<td>supermaximal repeat</td>
</tr>
</tbody>
</table>

Tabelle 4.4: Die Datensätze der DBT *admin.vmatch_seqtypes*

4.2.1.5 admin.vmatch_param_seqtypes

Mittels der beiden Attribute

- *param_id* (smallint)
• \textit{sequence_type} (varchar(20))

wird jedem \textit{sequence_type} - Attributwert eine (über die Klausel unique) eindeutige Nummer (\textit{param_id}) zugeordnet. Der Primärschlüssel setzt sich aus diesen beiden Attributen zusammen und \textit{sequence_type} ist zudem Fremdschlüssel zur übergeordneten DBT \textit{admin.seqtypes}. Tabelle 4.5 zeigt ein Beispiel an, dass als Grundlage für nachfolgende Ausführungen dient.

Auch hier setzt ein Trigger die Lückenlosigkeit von \textit{param_id} und die Kleinschreibung von \textit{sequence_type} durch. Im letzteren Fall kann u. U. durch den konvertierten Attributwert des Fremdschlüssels eine Fehlermeldung vermieden und darüber hinaus der Datensatz erfolgreich eingefügt werden.

\begin{center}
\begin{tabular}{|c|c|c|c|}
\hline
\textit{param_id} & \textit{sequence_type} & \textit{param_id} & \textit{sequence_type} \\
\hline
1 & repeat & 4 & palindrome \\
2 & repeat & 5 & supermaximal repeat \\
3 & palindrome & 6 & tandem \\
\hline
\end{tabular}
\end{center}

Tabelle 4.5: Die Datensätze der DBT \textit{admin.vmatch_param_seqtypes}

4.2.1.6 \textit{admin.repeat_id} / \textit{admin.palindrome_id}

Für diese beiden DBT werden die vorangegangenen Überlegungen zum Zwecke der Datenintegrität ausgenutzt und umgesetzt. Über die vier Attribute

• \textit{min_length} (integer)
• \textit{mismatch_type} (char(1))
• \textit{max_mismatches} (smallint)
• \textit{min_seed_length} (integer)

werden die in Kapitel 3 auf Seite 21 verwendeten VMATCH - Parameter eingetragen.

Die beiden Attribute

• \textit{param_id} (smallint)
• \textit{sequence_type} (varchar(20))

bilden zusammen den Primärschlüssel und sind zugleich Fremdschlüssel zur übergeordneten DBT \textit{admin.vmatch_param_seqtypes}. \textit{sequence_type} ist in \textit{admin.repeat_id} (\textit{admin.palindrome_id}) ausschließlich der Attributwert ‘repeat’ (‘palindrome’) zulässig. Damit

\begin{center}
\begin{tabular}{|c|c|c|c|c|c|}
\hline
\textit{param_id} & \textit{sequence_type} & \textit{min_len} & \textit{mismatch_type} & \textit{max_mismatches} & \textit{min_seed_length} \\
\hline
1 & repeat & 100 & hamming & 1 & 50 \\
2 & repeat & 200 & hamming & 10 & 40 \\
\hline
\end{tabular}
\end{center}

Tabelle 4.6: Die Datensätze der DBT \textit{admin.repeat_pid}
KAPITEL 4. LOGISCHER ENTWURF DER DB SYNTENY

<table>
<thead>
<tr>
<th>param_id</th>
<th>sequence_type</th>
<th>min_len</th>
<th>mismatch_type</th>
<th>max_mismatches</th>
<th>min_seed_length</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>palindrome</td>
<td>100</td>
<td>hamming</td>
<td>1</td>
<td>50</td>
</tr>
<tr>
<td>4</td>
<td>palindrome</td>
<td>200</td>
<td>hamming</td>
<td>10</td>
<td>40</td>
</tr>
</tbody>
</table>

Tabelle 4.7: Die Datensätze der DBT \textit{admin.palindrome}\_pid

wird erreicht, dass alle \textit{param_id} - Attributwerte mit dem \textit{sequence_type} 'repeat' ('palindrome') in \textit{admin.vmatch\_param\_seqtypes} exklusiv der DBT \textit{admin.repeat\_id} (\textit{admin.palindrome\_id}) gehören (s. u.). Die Tabellen 4.6 auf Seite 37 und 4.7 geben jeweils die beiden Datensätze für die verwendeten VMATCH - Parameter der Repeatberechnungen bzw. Palindromberechnungen an.

Werden andere VMATCH - Parameter verwendet, ist das Schema für \textit{admin.repeat\_id} respektive \textit{admin.palindrome\_id} entsprechend anzupassen. Würde man in diesen beiden DBT das Attribut \textit{sequence_type} mit fixem Wert weglassen, kann bspw. eine Eintragung der \textit{param_id} Werte 1 oder 2 in \textit{admin.palindromes} (so) nicht verhindert werden. Die Folge wäre eine Verletzung der Datenintegrität. Beide DBT bedienen sich eines eigenen Triggers für \textit{mismatch_type} und \textit{sequence_type}.

4.2.1.7 \textit{admin.supermaximal}\_repeat\_id / \textit{admin.tandem}\_id

Weil VMATCH für diese zwei Sequenzklassen keine fehlertoleranten Alignierungen im Sinne von Mispairs oder Gaps zulässt (s. Abschnitt 3 auf Seite 21), entfallen folglich im Vergleich zu \textit{admin.repeat\_id} und \textit{admin.palindrome\_id} die Attribute

- \textit{mismatch_type} (char(1))
- \textit{max_mismatches} (smallint)
- \textit{min_seed_length} (integer)

Von den verbleibenden drei Attributen

- \textit{param_id} (smallint)
- \textit{sequence_type} (char(20))
- \textit{min_length} (integer)

bilden die zwei erstgenannten jeweils den Primärschlüssel. Wie man aus den Tabellen 4.8 und 4.9 auf Seite 39 erkennen kann, unterliegt hier \textit{sequence_type} ebenfalls den oben beschriebenen Restriktionen einer unveränderlichen Wertzuweisung.

Das Schema \textit{admin} kann ggf. relativ einfach auf andere Sequenzklassen wie die bereits erwähnten \textit{mums} erweitert werden.

Ein Trigger konvertiert \textit{sequence_type} in Kleinbuchstaben.

<table>
<thead>
<tr>
<th>param_id</th>
<th>sequence_type</th>
<th>min_len</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>supermaximal</td>
<td>repeat</td>
</tr>
</tbody>
</table>

Tabelle 4.8: Auszug aus der DBT \textit{admin.supermaximal}\_repeat\_id

38
KAPITEL 4. LOGISCHER ENTWURF DER DB SYNTENY

<table>
<thead>
<tr>
<th>param_id</th>
<th>sequence_type</th>
<th>min_len</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>tandem</td>
<td>75</td>
</tr>
</tbody>
</table>

Tabelle 4.9: Auszug aus der DBT admin.tandem_id

4.2.1.8 admin.dna_carrier

Hier wird jedem DNA-Erbräger (s. Tabelle 4.10) wie folgt eine ID zugeordnet:

- `carrier_id` (smallint)
- `species_id` (smallint)
- `version_id` (smallint)
- `carrier_name` (char(2))

sind die vier Attribute, von denen `carrier_id` der Primärschlüssel ist. `species_id` und `version_id` sind Fremdschlüssel zu `admin.species`, `version` ist die biologisch eindeutige Bezeichnung des DNA-Erbrägers (z. B. '1', '2', ..., 'X', 'Y'). Der ebenfalls eindeutigen Attributkombination (`species_id` (2 Byte), `version_id` (2 Byte), `carrier_name` (2 Byte)) wird eine `carrier_id` zugeordnet. Da `carrier_id` (2 Byte) anstelle der anderen drei Attribute in jedem der ca. 165 Mio. Duplikattupel angegeben werden muss, spart dies möglicherweise (erheblichen) Speicherplatz und beschleunigt ggf. Abfragen durch dichtere Packungsdichte der Tupel in Seiten (s. Abschnitt 5.1.3 auf Seite 51).

Zur Verhinderung von biologisch unzulässigen Einträgen ('Z') bezüglich `carrier_name`, wird durch einen Trigger sichergestellt, dass nur Werte '1', '2', ..., '98', '99', 'X', 'Y' akzeptiert werden. Auch wenn Spezies mit mehr als 99 Chromosomen existieren, sollte dies für haploide Chromosomensätze aber i. A. ausreichend sein und kann bei Bedarf auf einfache Weise eingeschränkt oder erweitert werden. In diesem Zusammenhang werden durch den Trigger führende Nullen aus Konsistenzgründen bei zulässigen Eingaben ('01', '02', ..., '09') entfernt ('1', '2', ..., '9').

<table>
<thead>
<tr>
<th>carrier_id</th>
<th>species_id</th>
<th>version_id</th>
<th>carrier_name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>'1'</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>2</td>
<td>'22'</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>2</td>
<td>'X'</td>
</tr>
<tr>
<td>24</td>
<td>1</td>
<td>2</td>
<td>'Y'</td>
</tr>
<tr>
<td>25</td>
<td>3</td>
<td>4</td>
<td>'1'</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Tabelle 4.10: Auszug aus der DBT admin.dna_carrier

4.2.1.9 admin_pal_cid_pid / admin_rep_cid_pid / admin_supmax_cid_pid / admin.tandem_cid_pid

Mittels der beiden Attribute

- `carrier_id` (smallint)
KAPITEL 4. LOGISCHER ENTWURF DER DB SYNTENY

• **param_id** (smallint)

wird festgehalten, welche Sequenzklassenberechnungen (über **param_id**) für welche DNA-Erbträger (über **carrier_id**) vorliegen. Beide Attribute zusammen ergeben den Primärschlüssel und sind jeweils Fremdschlüssel: **carrier_id** zu **admin.dna_carrier** und **param_id** zu genau einer in Abschnitt 4.2.1.6 auf Seite 37 bzw. 4.2.1.7 auf Seite 38 angegebenen Tabelle. Über diese beiden Attribute kann jedes Duplikattupel genau einem VMATCH-Lauf und einem bestimmten DNA-Erbträger einer Spezies in einem bestimmten Versionsstand zugeordnet werden. Alternativ ist es denkbar, einem separaten Primärschlüssel (**new_pk**) zu definieren, der dann die Zuordnung in einem Duplikattupel herstellt. Dem geringen Vorteil einer Platzersparnis von 4 Byte (**carrier_id** und **param_id**) - 2 Byte (**new_pk**) = 2 Byte pro Duplikattupel stehen alles in allem deutlich gravierendere Nachteile gegenüber:

• Informationen, welches Duplikat zu welchen VMATCH-Lauf und DNA-Erbträger gehört, ist dann nur noch über eine Verknüpfung (Join) mit dieser DBT durchführbar.

• u.U. können (insbesondere für laufzeitkritische Abfragen) nicht die 'bestmöglichen' Indizes realisiert werden, z.B. wenn die Anzahl der Duplikate nach **carrier_id** und / oder (**carrier_id**, **param_id**) gruppiert werden soll.

Diese Erwägungen belassen es bei dem oben beschriebenen Schema für diese DBT.

Trigger sind hier nicht erforderlich.

4.2.2 Sequenzkomponente

Den Datenbankobjekten der Sequenzkomponente wird der Schemaname **sequence** vorangestellt. Der Komponente zugehörig sind drei DBT, von denen die erste (**sequence.dna_sequences**) die DNA-Sequenzen beinhaltet, die zweite (**sequence.dna_sequence_gaps**) - falls vorhanden - Koordinaten von unbestimmten Basenfolgen, sowie eine DBT (**sequence.cid_seq_id**), die eine Relation zur DBT **admin.dna_carrier** der Administrationskomponente herstellt und daher auch den Schemanamen **admin** tragen könnte. Abbildung 4.4 gibt diesen Sachverhalt bildlich wieder.

Die hier bereitgestellten Sequenzinformationen werden hauptsächlich für die Datenaufbereitung genutzt.

4.2.2.1 **sequence.dna_sequences**

Jene DBT mit den Attributen

• **sequence_id** (smallint)
KAPITEL 4. LOGISCHER ENTWURF DER DB SYNTENY

- dna_sequence (CLOB(250M))

speichert über den Primärschlüssel sequence_id im Feld dna_sequence die DNA-Sequenz als CLOB ab. Darüber hinaus ist der Primärschlüssel auch Fremdschlüssel zu sequence.cid_seq_id.

Die DNA-Sequenzen sind mittels IMPORT ohne nennenswerten Nachteil gegenüber LOAD in sequence.dna_sequences eingriffbar. Da IMPORT Trigger sofern vorhanden - auslöst, bot sich die Realisierung folgender Überlegungen an:


Diese Datenaufbereitung sollte ein SQL PL Trigger übernehmen, der gleichzeitig die Lückenlosigkeit von sequence_id sicherstellt. Bei dieser Gelegenheit könnte nach erfolgreicher Eintragung ein weiterer Trigger ausgelöst werden, der die aktuelle DNA-Sequenz auf Gaps untersucht und gefundene Gapkoordinaten einschließlich sequence_id in die DBT sequence.sequence_gaps einträgt.

Der dazu erstellte Trigger sequence.transformfasta ist ein BEFORE - INSERT - TRIGGER. Mit den vorhandenen und benötigten skalaren SQL-Funktionen wie TRANSLATE respektive UPPER (alias UCASE), REPLACE, oder LOCATE [IBM IV] konnte aufgrund einer Stringlängenbegrenzung auf rund 1 Mio. Zeichen die Datenaufbereitung nur schrittweise durchgeführt werden. Die holende SQL-Substringfunktion SUBSTR lässt sogar nur 254 Zeichen zu, sodass das Humanchromosom 1 in fast 1 Mio Substrings zerlegt werden muss, mit der Folge eines erheblichen Overheads in Form von Hilfsvariablen und Hilfszuweisungen. Mittels SUBSTR wird ein DNA-String herausgenommen und in zwei Hilfsvariablen zweiseitengespichert. In der einen Hilfsvariable werden mittels TRANSLATE alle 'a', 'c', 'g', 't', 'n', 'A', 'C', 'G', 'T' durch 'N' ersetzt, um anschließend alle 'N' und durch CHR(10) gefundenen Zeilenumbrüche mittels REPLACE herauszuschneiden. Bleibt ein nicht-leerer String übrig, existieren entweder noch andere IUPAC - Basen oder Fremdzeichen. Tritt ausschließlich der erste Fall auf, so werden alle IUPAC - Basen in der zweiten Hilfsvariable durch 'N' generalisiert und abschließend die modifizierte DNA-Sequenz in die DBT sequence.sequence_dna_sequences inklusive sequence_id eingetragen. Im zweiten Fall wird ohne Eintragung der Einfügeprozess mit einer Fehlermeldung abgebrochen.


3s. Tabelle 6.1 auf Seite 62
Eine Schemänderung mit der Zerlegung der zu importierenden DNA- Sequenz und der Annotation der Anfangs- und Endpositionen jeder einzelnen Subsequenz wird verworfen, weil dies zu einer deutlichen Erhöhung der Komplexität von Abfragen mit Sequenzbezug führt. Vorstellbar ist aber eine externe (C, Java, Perl, ...) Datenbankanwendung, die die DNA-Sequenz aus der FASTA-Datei einliest und mittels embedded SQL in sequence.dna_sequences inseriert.

Aus diesem Grund wurde sequence.get_dna_gaps ersatzlos gestrichen und sequence.transformfasta auf die Überprüfung der Lückenlosigkeit von sequence_id reduziert und umbenannt. Die Transformation der DNA-Sequenzen sowie die Bestimmung der Gapkoordinaten wurde kategorisch von einem C-Programm übernommen, welches das FASTA-DNA-Flatfile als Eingabe verwendet und die Gapkoordinaten samt sequence_id (als Parameter übergeben) im DEL-Format von DB2 heraussschreibt.

4.2.2.2 sequence.dna_sequence_gaps

Über die vier Attribute

- sequence_id (smallint)
- gap_start (integer)
- gap_end (integer)
- gap_length (integer)


4.2.2.3 sequence.cid_seq_id

Mit Hilfe der beiden Attribute

- carrier_id (smallint)
- sequence_id (smallint)


42
KAPITEL 4. LOGISCHER ENTWURF DER DB SYNTENY

geschieht über carrier_id und sequence_id. Bei mehreren GB Speicherplatzbedarf pro Genom (und Version), sind zum einen beachtliche Speicherplatz einsparungen erreichbar und zum anderen redundante Sequenzeinträge vermeidbar.

4.2.3 Duplikatkomponente

Die Duplikatkomponente verwaltet alle Duplikate und zeichnet sich durch den Schemaname duplicates aus. Die DBT duplicates.repeats native (duplicates.palindromes native) werden samt ihrer Unterschiede zu duplicates.repeats (duplicates.palindromes) in Abschnitt 4.2.4 auf Seite 43 kurz beschrieben.

4.2.3.1 duplicates.palindromes / duplicates.repeats / duplicates.supermax_repeats / duplicates.tandems

Im Folgenden werden sukzessive die gemeinsamen Attribute der obigen sechs DBT der Duplikatkomponente vorgestellt. Auf spezifischere Attribute wird gesondert eingegangen. Die vier Attribute

- pal_id (duplicates.palindromes) (integer)
- repeat_id (duplicates.repeats) (integer)
- supermax_id (duplicates.supermax_repeats) (integer)
- tandem_id (duplicates.tandems) (integer)

sind jeweils die Primärschlüssel zu den in Klammern angegebenen DBT. Sie werden mittels der Klausel generated always as identity erzeugt. Die bereits angesprochene Lücke losigkeit der Attributwerte ist hier irrelevant, da einerseits die Daten über LOAD eingefügt werden, das keine Trigger auslöst und andererseits der Wert des Primärschlüssels keinen assoziativen Bezug zum Tupel hat.

Die beiden Attribute

- carrier_id (smallint)
- param_id (smallint)

enthalten den zusammengesetzten Fremdschlüssel zur entsprechenden übergeordneten DBT (s. Abschnitt 4.2.1.9 auf Seite 39). Falls bspw. <tcid> und <tpid> Attributwerte aus einem Datensatz von duplicates.tandems sind, so müssen beide zusammen auch als Attributwerte (irgend-) eines Datensatzes in admin.tandem_cid_pid vorkommen. Dadurch wird jeder Duplikatdatensatz (hier) genau einem Chromosom (inklusive Version) und VMATCH - Lauf eindeutig zugeordnet. Die folgenden vier Attribute

- startpos1 (integer)
- endpos1 (integer)
- startpos2 (integer)
- endpos2 (integer)

sind die Koordinaten der beiden Duplikatarme. Ohne Beschränkung der Allgemeinheit gelte für jedes Tupel startpos1 < startpos2.

Weil VMATCH bei Tandems zwischen beiden Tandemarmen kein Gap zulässt, gilt in diesem speziellen Fall
KAPITEL 4. LOGISCHER ENTWURF DER DB SYNTENY

• \( \text{startpos2} = \text{endpos1} + 1 \)

• \( \text{endpos2} = 2 \cdot \text{endpos1} + 1 - \text{startpos1} \)

und wird über die generated always Klausel erzwungen, sodass hier beim Füllen der Tabelle nur startpos1 und endpos1 angegeben werden dürfen. Diese Vorgehensweise bietet einen vereinfachten Zugriff auf beide Tandemarme und ist schemakonsistent zu allen anderen Duplikaten. Bei insgesamt nur einigen tausend Tandems mit geringem Platzbedarf pro Tupel erübrigt sich jede Diskussion wegen erhöhten Speicherplatzanforderungen.

Das Attribut für die Duplikatarmlänge

• \( \text{length} \) (integer)

wird ebenfalls über die generated always Klausel nach der Formel \( 1 + \text{endpos1} - \text{startpos1} \) berechnet. Die Duplikatlänge ist zum einen für nachfolgende Datenverarbeitungen notwendig. Eine spürbare Beschleunigung der Abfrage von Duplikatlängen ermöglichen geeignete Indizes unter Einbeziehung von length. Zum anderen ist es aufwendiger, die Länge stets mittels SQL 'on - the - fly' auszurechnen. Nachteilig ist der signifikante Mehrverbrauch an Speicherplatz.

Das Attribut

• \( \text{eval} \) (double / 8 Byte)

wird von VMATCH übernommen.

Für die Beantwortung einer biologischen Fragestellung (Syntenieuntersuchung) dienen die fünf Attribute

• \( \text{startpos1} \_\text{func}_\text{id} \) (integer)

• \( \text{endpos1} \_\text{func}_\text{id} \) (integer)

• \( \text{startpos2} \_\text{func}_\text{id} \) (integer)

• \( \text{endpos1} \_\text{func}_\text{id} \) (integer)

• \( \text{num} \_\text{func} \) (smallint)


In \( \text{num} \_\text{func} \) wird wiederum unter Verwendung der generated always Klausel die Anzahl der in Genbereichen liegenden (Duplikatarm-)Koordinaten eingetragen (Werte von 0 bis 4), sobald die vorherigen vier Attribute keine NULL - Werte mehr besitzen. Das automatische Berechnen von \( \text{num} \_\text{func} \) aus den anderen vier Attributen soll die Datenintegration wahren. \( \text{num} \_\text{func} \) enthält genau dann keinen Nullwert, wenn alle vier anderen Attribute keinen Nullwert (mehr) enthalten. Über das Anlegen eines sinnvollen Index erfolgt ein effizienter Zugriff auf hier interessierende \( \text{num} \_\text{func} \) Attributwerte.

Aus semantischen Beweggründen ist prinziell eine für diese fünf Attribute separate DBT in der Ergebniskomponente anzulegen, da es sich bereits um eine Datenauswertung handelt. Die Zuordnung dieser fünf Attribute erfolgt dann über ein sechstes Attribut, das gleichzeitig Primärschlüssel und Fremdschlüssel zur entsprechenden DBT in der Duplikatkomponente ist:

Das Attribut

- `dia_value` (integer)

ist ein mittels `generated always` Klausel berechnetes Attribut für die Datenaufbereitung und -filterung mit repeat- und palindromspezifischer Berechnungsformel (s. u. a. Abschnitt 6.2.2 auf Seite 66).

### 4.2.3.2 `duplicates.palindromes` / `duplicates.repeats`

Die folgenden beiden Attribute

- `mismatches` (smallint)
- `prcnt_ident` (decimal(6,3) / 4 Byte)

geben die Anzahl der Mismatches (`mismatches`) an, sowie die über die `generated always` Klausel berechnete prozentuale Übereinstimmung (`prcnt_ident`) der Repeat- bzw. Palindromarme. Die zwei Attribute

- `red_id` (integer)
- `native_cluster_id` (integer)

sind auch für die Datenaufbereitung vorgesehen. In Abhängigkeit von `dia_value` - Attributwerten werden redundante, d. h. sich vollständig überlappende Repeats und Palindrome über bestimmte `red_id` Attributwerte zunächst markiert (s. Abschnitt 6.2.2.1 auf Seite 66), um sie dann zielgerichtet zu löschen.

Repeats (Palindrome) die sich auf besondere Art nach einem `dia_value` - Kriterium teilweise überlappen, werden mittels einer gleichen und eindeutigen Clusternummer, `native_cluster_id` markiert, um anschließend zu einem Repeat (Palindrom) zusammengefasst und anschließend entweder nach `duplicates.repeats native` bzw. `duplicates.palindromes native` verschoben zu werden. Diesem Vorgang widmen sich die Abschnitte 6.2.2.2 auf Seite 71 und 6.2.2.3 auf Seite 75.

Das Attribut

- `overlap_id` (integer)

markiert ggf. Duplikate aus verschiedenen VMATCH - Läufen als 'Subduplikat' genau dann, wenn sie sich auf dem gleichen Erbträger befinden und eben eine DNA - Subsequenz eines längeren Duplikates sind. Es sollte nicht überraschen, wenn Duplikate mit der Mindestlänge 100 aufgrund ihrer stringenteren VMATCH - Restriktionen 'nur' eine DNA - Subsequenz eines längeren Duplikates mit Mindestlänge 200 ist. Für eine graphische Ausgabe beider Duplikate im Webbrowser sind nur die Koordinaten des längeren Duplikates notwendig, da es das kürzere Duplikat quasi 'übermalt'. Über dieses Attribut geschieht dann eine Duplikatfilterung. Abschnitt 6.2.2.5 auf Seite 81 beschreibt die Durchführung des Markierens.

Von
wird nur für den Fall Gebrauch gemacht, wenn sich beim physischen Datenbankentwurf herausstellen sollte, dass MDC\(^4\)-DBT vorteilhaft sind (s. Kapitel 5 ab Seite 48). Dann generalisieren diese vier Attribute die Attributwerte ihrer Positionspendants, um Duplikate mit gleichen MDC-Attributwerten zu gruppieren.

4.2.4 Partiell verwendete Attribute
Wie in Kapitel 6 auf Seite 61 erläutert wird, ist eine Datenaufbereitung der Repeats und Palindrome notwendig. Die hier genannten Attribute werden ausführlicher in dem angeführten Kapitel erläutert. Wegen der Datenaufbereitung enthalten \textit{duplicates.repeats} und \textit{duplicates.palindromes} die Attribute \textit{redundance_id} und \textit{native_cluster_id}. Die Tupel der DBT \textit{duplicates.repeats\textsubscript{native}} und \textit{duplicates.palindromes\textsubscript{native}} sind die Ergebnisse diverser Datenaufbereitungsschritte und benötigen diese beiden Attribute nicht. \textit{num_parts} ist ein Attribut, welches nur \textit{duplicates.repeats\textsubscript{native}} und \textit{duplicates.palindromes\textsubscript{native}} wegen der Datenaufbereitung besitzen. Das Attribut \textit{evalue} wird von VMATCH berechnet und hier übernommen. Die Neuberechnung von \textit{evalue} für \textit{duplicates.repeats\textsubscript{native}} und \textit{duplicates.palindromes\textsubscript{native}} ist nur mit unverhältnismäßigem Aufwand möglich, sodass dieses letzteren beiden DBT fehlt.

4.2.5 Ergebniskomponente
Für einige Untersuchungen und Auswertungen wurden bei Bedarf weitere DBT erzeugt, die unter diversen Gesichtspunkten Daten aus anderen DBT von SYNTENY temporär zusammenfassten. Alle diese DBT, die nur zu Informationszwecken angelegt wurden und daher verzichtbar sind, werden durch den Schemamen\textit{results} gekennzeichnet. Daher erübrigen sich detailliertere Angaben zu den DBT von \textit{results}.

4.3 Trigger
Trigger wurden im Kontext dieser Diplomarbeit aus verschiedenen Gründen eingesetzt. Zu beachten ist u. a., dass z. B. \textit{LOAD} - Operationen keine Trigger auslösen. U. U. sind dann andere Strategien zur Wahrung der Datenintegrität in Erwägung zu ziehen.
Die meisten der im Folgenden vorzustellenden Identifier werden durch die DDL - Klauseln \textit{generated by default as identity} bzw. \textit{generated always as identity} erzeugt. Ihre konkreten Werte sind dabei inerhoblich, wichtig ist jedoch ihre Eindeutigkeit. I. A. sind die derart erzeugten Identifier 'lückenlos': Unter der Abfolge von bestimmten Datenbankoperationen sind lückenbehafte Identifier - Attributwerte (angenommen: 1; 2; 5; 7) einer DBT erzeugbar (fehlend: 3; 4; 6)\[B04\]. Daraus ergeben sich keine ernsthaften Folgen, aber die Feststellung solcher Lücken durch Nutzer kann diese verwirren und unnötige Fragen aufwerfen. Der restriktive Einsatz von Triggern zur Vorbeugung dieses Problems macht obige DDL - Klauseln obsolet. In (eher theoretischen) Ausnahmefällen können Trigger die Lückenlosigkeit zwar auch nicht garantieren [B04], aber mit einer exklusiven Datenbankverbindung lässt sich die Wahrscheinlichkeit eines solchen Ausnahmefalles drastisch reduzieren.

\[multidimensional\ clustering\]
DROP TRIGGER <SCHEMANAME>.<TRIGGERNAME>@
CREATE TRIGGER <SCHEMANAME>.<TRIGGERNAME>
NO CASCADE BEFORE INSERT ON <SCHEMANAME>.<TABLENAME> REFERENCING NEW AS N
FOR EACH ROW MODE DB2SQL
BEGIN ATOMIC
DECLARE MAX_VALUE INTEGER;
SET MAX_VALUE = (SELECT MAX(<ID_ATTRIBUTE_VALUE>) FROM <SCHEMANAME>.<TABLENAME>);

IF (MAX_VALUE IS NULL) THEN
  IF (N.<ID_ATTRIBUTE_VALUE> <> 1) THEN
    SIGNAL SQLSTATE '<SQLSTATE>'
    SET MESSAGE_TEXT = 'NO VALID ID_ATTRIBUTE_VALUE';
    END IF;
  ELSE
    IF (N.<ID_ATTRIBUTE_VALUE> <> (MAX_VALUE + 1)) THEN
      SIGNAL SQLSTATE '<SQLSTATE>'
      SET MESSAGE_TEXT = 'NO VALID ID_ATTRIBUTE_VALUE';
    END IF;
  END IF;
ELSE
  IF (N.<ID_ATTRIBUTE_VALUE> <> (MAX_VALUE + 1)) THEN
    SIGNAL SQLSTATE '<SQLSTATE>'
    SET MESSAGE_TEXT = 'NO VALID ID_ATTRIBUTE_VALUE';
  END IF;
ENDIF;
SET N.<NAME> = LOWER(N.<NAME>);
END@
Kapitel 5

Überlegungen zum physischen Datenbankentwurf

Nach der Vorstellung des logischen Schemas der DB SYNTENY in Kapitel 4 auf Seite 31, steht hier die physische Umsetzung des Schemas im Vordergrund. Die Abschnitte 5.1 und 5.4 auf Seite 53 befassen sich mit der Konfiguration der Tabellenbereiche, der Abschnitt 5.2 auf Seite 51 mit den Anforderungen an den Speicherbedarf der Protokollierung und der darauf folgende Abschnitt 5.3 auf Seite 52 mit der Einrichtung der Pufferpools. Der vorletzte Abschnitt 5.5 auf Seite 59 dieses Kapitels fasst hauptsächlich die Dateneinspeisung zusammen und wird von Abschnitt 5.6 auf Seite 60, der Kapitelzusammenfassung, abgerundet.


Die Überlegungen zum physischen Datenbankentwurf basieren - sofern nicht anders angegeben - auf [IBM] und [M04].

5.1 Tabellenbereiche

Ein Tabellenbereich (table space) ist ein Bereich zum Speichern von DBT. Bei der Erstellung einer DBT mittels create table <schemaname>.<tablename> können bestimmte Datenbankobjekte wie Indizes und LOB - Daten von den übrigen Tabellendaten getrennt gespeichert werden. Es wird zwischen Tabellenbereichen unterschieden, die vom Betriebssystem (SMS -

\(^1\)länge insgesamt nur ein Bruchteil der Datenmenge vor, ist eine Performanzsteigerung u. U. kaum wahrnehmbar und deswegen möglicherweise verzichtbar
KAPITEL 5. ÜBERLEGUNGEN ZUM PHYSISCHEN DATENBANKENTWURF

system managed space) bzw. der DB (DMS - database managed space) verwaltet und über Behälter (container) in Form von Dateien, Verzeichnissen oder Laufwerken diesen zugeordnet werden.

Ein SMS sollte den system catalog Tabellenbereich aufgrund dynamischer Speicherplatzänderungen verwendet werden (Voreinstellung). DMS dagegen für alle anderen Tabellenbereiche (Daten, [s. Abschnitt 5.1.1], LOB's [s. Abschnitt 5.1.2 auf Seite 50] Indizes [s. Abschnitt 5.1.3 auf Seite 51] zu einer DBT die Zugriffszeiten verbessert und zu einer 5 - 10 prozentigen Leistungsverbesserung führen kann.

Es ist daher empfehlenswert, die durch create database SYNTENY erzeugten SMS - Tabellenbereiche TEMPSPACE1 und USERSPACE zu ersetzen.

Für die DB SYNTENY standen zwei Festplatten mit jeweils rund 34 GB Speicherkapazität zur Verfügung.

Auf diese beiden Festplatten - nachfolgend mit vol1 und vol2 bezeichnet - wurden die DMS - Tabellenbereiche wie in Tabelle 5.1 angegeben, aufgeteilt. Die Tabellenbereichscontainer sind Dateien. Die beiden Verzeichnispfade der Festplatten seien /db/vol1/ und /db/vol2/.

<table>
<thead>
<tr>
<th>Festplatte</th>
<th>Tabellenbereich</th>
<th>Name</th>
<th>init. Dateigröße</th>
<th>init. Containeranzahl</th>
</tr>
</thead>
<tbody>
<tr>
<td>vol1</td>
<td>Daten</td>
<td>SYN_TS_DATA</td>
<td>1024 Megabyte</td>
<td>24</td>
</tr>
<tr>
<td>vol1</td>
<td>Long</td>
<td>SYN_TS_LONG</td>
<td>1024 Megabyte</td>
<td>8</td>
</tr>
<tr>
<td>vol2</td>
<td>Index</td>
<td>SYN_TS_IDX</td>
<td>1024 Megabyte</td>
<td>10</td>
</tr>
<tr>
<td>vol2</td>
<td>Temp</td>
<td>SYN_TS_TEMP</td>
<td>1024 Megabyte</td>
<td>8</td>
</tr>
</tbody>
</table>

Tabelle 5.1: Aufteilung der DMS - Tabellenbereiche

Während vol1 fast vollständig belegt ist (24 + 8 Dateien á 1 GB = 32 GB), ist vol2 rund zur Hälfte (10 + 8 Dateien á 1 GB = 18 GB) belegt.

5.1.1 Daten

Für die Speicherung der rund 165 Mio. Duplikate sind Abschätzungen zur Speicherplatzbelegung ratsam. Schon allein das Abspeichern der Koordinaten \((i, j)\) und \((k, l)\) der Duplikatarme \(S_1[i..j]\) und \(S_2[k..l]\) mit vier Byte (integer) pro Koordinate verschlingt 165.000.000 • 16 Byte = 2,64 GB. Jedes zusätzliche integer - Attribut ergo 0,66 GB. Werden die verwendeten 49 DNA - Sequenzen in eine DBT abgelegt, so sind dafür schon schätzungsweise 6 GB Speicherplatz erforderlich.

Pro Datensatz, der die gewünschten Attribute für jedes Duplikat enthält, werden bei Wahl einer nicht - MDC - DBT (s. Abschnitt 4.2.3 auf Seite 43) 76 Byte veranschlagt, bei Verwendung einer MDC - DBT 92 Byte.

Datensätze in DB2 - DBT’s werden in Seiten (pages) gruppiert, die unterschiedlich groß sein können: 4 KB (4096 Bytes), 8 KB (8192 Bytes), 16 KB (16384 Bytes) oder 32 KB (32768 Bytes). Die Festlegung auf einen dieser vier Werte erfolgt über einen Parameter, der Seitengröße (pagesize).

Die folgenden Berechnungen gehen von einer Verwendung von 4 KB - Seiten (Standardfestlegung) aus:

- 76 Byte pro Tupel
  - Berechnung der Datensätze pro Seite:

49
KAPITEL 5. ÜBERLEGUNGEN ZUM PHYSISCHEN DATENBANKENTWURF

– $\left\lfloor \frac{4028 \text{ Byte}}{\text{Tupelgroesse (Byte) + 10 \text{ Byte}}} \right\rfloor$ = Anzahl Tupel pro Seite

– $\left\lfloor \frac{4028 \text{ Byte}}{76 \text{ Byte} + 10 \text{ Byte}} \right\rfloor$ = 46 Tupel pro Seite

– Berechnung der benötigten Seiten:

– $\left\lceil \frac{\text{Anzahl Tupel}}{\text{Anzahl Tupel pro Seite}} \right\rceil$ = Anzahl Seiten

– $\left\lceil \frac{165.000.000}{46} \right\rceil$ = 3.586.957 Seiten

– Berechnung des benötigten Speicherplatzes:

– 3.586.957 \cdot 4096 \approx 14,7 \text{ GB}

• 92 Byte pro Tupel

– Berechnung der Datensätze pro Seite:

– $\left\lfloor \frac{4028 \text{ Byte}}{\text{Tupelgroesse (Byte) + 10 \text{ Byte}}} \right\rfloor$ = Anzahl Tupel pro Seite

– $\left\lfloor \frac{4028 \text{ Byte}}{92 \text{ Byte} + 10 \text{ Byte}} \right\rfloor$ = 39 Tupel pro Seite

– Berechnung der benötigten Seite:

– $\left\lceil \frac{\text{Anzahl Tupel}}{\text{Anzahl Tupel pro Speicherseite}} \right\rceil$ = Anzahl Seiten

– $\left\lceil \frac{165.000.000}{39} \right\rceil$ = 4.230.770 Seiten

– Berechnung des benötigten Speicherplatzes:

– 4.230.770 \cdot 4096 \approx 17,4 \text{ GB}

5.1.2 LOBs
Der Speicherbedarf für die DNA-Sequenzen in Form von CLOB’s lässt sich ebenfalls anhand von Faustformeln errechnen. In der Praxis hat sich jedoch herausgestellt, dass in etwa der Speicherbedarf pro Chromosom mit der Dateigröße übereinstimmt. Zur Sicherheit werden darauf etwa 10 - 20% aufgeschlagen. Bei Bedarf lässt sich anschließend der LONG-Tabellenbereich mittels alter tablespace SYN_TS_LONG und resize verkleinern.
KAPITEL 5. ÜBERLEGUNGEN ZUM PHYSGISCHEN DATENBANKENTWURF

5.1.3 Indexes
Die Festlegung von geeigneten Attributen für die Zusammensetzung eines Index ist u. U. nicht trivial. Intuitiv wird man bspw. wohl einen nicht-eindeutigen Index (\texttt{species\_id} (smallint / 2 Byte), \texttt{carrier\_id} (smallint / 2 Byte), \texttt{param\_id} (smallint / 2 Byte)) festlegen, um Duplikatdaten einem bestimmten VMATCH-Lauf (siehe bspw. Tabelle 4.6 auf Seite 37) bezüglich eines bestimmten Erbträgers (\texttt{carrier\_id}) und Spezies (\texttt{species\_id}) zuzuordnen.

Der erforderliche Speicherplatz kann dann wie folgt abgeschätzt werden:

- \((\emptyset \times \text{Indexkeysizes Byte} + 9 \times \text{Byte}) \times \text{Anzahl Tupel} \times 2 \times \text{Systemaufwand})\)

- \((2 + 2 + 2) \times \text{Byte} + 9 \times \text{Byte}) \times 165.000.000 \times 2 \approx 5,0 \text{ GB}\)

Auch wenn die verwendete Formel von eindeutigen Indizes ausgeht, und der veranschlagte Speicherplatz daher als zu groß abgeschätzt wird, verdeutlicht dieses Beispiel dennoch, dass mehrere Indizes viel Speicherplatz belegen können.

Hätte man anstelle der drei smallint-Datentypen integer-Datentypen verwendet, so wäre der Speicherbedarf um 40% auf ca. 7,0 GB gestiegen.

Erschwerend für die Speicherkalkulation wirkt sich einerseits die Anzahl der erstellten oder noch zu erstellenden Indizes aus, andererseits wurde in der Entwicklungsphase festgestellt, dass der Indexadvisor (\texttt{db2advis}) für ‘wichtige’/laufzeitkritische SQL-Queries teilweise intuitiv unerklärliche Indizes vorschlägt. Somit stellen die Anzahl der Indizes und die durchschnittliche Indexschlüsselgröße kaum kalkulierbare Größen für den Speicherplatzverbrauch von Indizes dar.

Der während der Indexerstellung maximal benötigte temporäre Speicherplatz kann durch folgende Formel abgeschätzt werden:

- \((\emptyset \times \text{Indexkeysizes Byte} + 9 \times \text{Byte}) \times \text{Anzahl Tupel} \times 3,2 \times \text{Systemaufwand})\)

Für obiges Beispiel wären demnach mindestens

- \((2 + 2 + 2) \times \text{Byte} + 9 \times \text{Byte}) \times 165.000.000 \times 3,2 \approx 8,0 \text{ GB}\)

temporärer Speicherplatz erforderlich.

5.2 Speicherbedarf für Protokolldateien
Der minimal zur Verfügung stehende Speicherplatz für die aktive Protokollkonfiguration berechnet sich bei Verwendung von 4 KB-Seiten nach der Formel:

- \((\text{logprimary} + \text{logsecond}) \times (\text{logfilsiz} + 2) \times 4096)\)

Die Werte für die Parameter \texttt{logprimary} und \texttt{logsecond} können über die Kommandozeile mittels

\texttt{db2 get db cfg}\)

eingesehen und mittels
KAPITEL 5. ÜBERLEGUNGEN ZUM PHYSISCHEN DATENBANKENTWURF

<table>
<thead>
<tr>
<th>LOG - Konfigurationsparameter</th>
<th>neuer Wert</th>
</tr>
</thead>
<tbody>
<tr>
<td>logbufsz</td>
<td>32.768</td>
</tr>
<tr>
<td>logfilsze</td>
<td>131.072</td>
</tr>
<tr>
<td>logprimary</td>
<td>16</td>
</tr>
<tr>
<td>logsecond</td>
<td>4</td>
</tr>
</tbody>
</table>

Tabelle 5.2: geänderte Protokollierungsparameter

```db2 update db cfg using <parameter> <new_parameter_value>```

geändert werden.

Aus Performanzgründen sollte die Protokollierung auf (mindestens) einer separaten Hochgeschwindigkeitsfestplatte erfolgen (`db2 update db cfg using newlogpath <new_path>`).

Tabelle 5.2 gibt die geänderten Werte für die DB SYNTENY an. Nach obiger Formel müssen daher mindestens

- \((16 + 2) \times (131.072 + 2) \times 4096\) Byte \(\approx 9.7\) GB

Speicherkapazität bereit stehen. Daher erfolgt die Protokollierung auf `vol2`.

5.3 Pufferpools


Als Ausgangsbasis empfiehlt es sich, jeweils einen Pufferpool für

- temporäre Tabellenbereiche (mittel)
- indexbezogene Tabellenbereiche (groß)
- datenbezogene Tabellenbereiche (groß)

anzulegen. Da für LOB’s keine spezifischen Pufferpools erstellt werden, können sie irgendwelchem Pufferpool zugewiesen werden.

Die Konfiguration der Pufferpools ist der wichtigste Einzelbereich der Optimierung und unterliegt u. U. stark leistungshemmenden Restriktionen: 32 Bit Betriebssysteme (bzw. 32 Bit

<table>
<thead>
<tr>
<th>Betriebssystem</th>
<th>max. shared memory</th>
</tr>
</thead>
<tbody>
<tr>
<td>AIX</td>
<td>1,75 GB</td>
</tr>
<tr>
<td>Linux</td>
<td>1,75 GB</td>
</tr>
<tr>
<td>Sun</td>
<td>3,35 GB</td>
</tr>
<tr>
<td>Windows</td>
<td>2,0 - 3,0 GB</td>
</tr>
</tbody>
</table>

Tabelle 5.3: 32 Bit Betriebssysteme und deren max. shared memory
KAPITEL 5. ÜBERLEGUNGEN ZUM PHYSISCHEN DATENBANKENTWURF

Instanzen) beschränken den shared memory wie folgt (Auswahl). Der shared memory setzt sich aus den in Tabelle 5.4 angegebenen DB - Komponenten zusammen. Zu den Wertebe- reichen s. [IBM II].

<table>
<thead>
<tr>
<th>Komponente</th>
<th>Konf.</th>
<th>Performanz-relevanz</th>
<th>zulässiger Wertebereich</th>
<th>Default</th>
<th>Maßeinheit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pufferpool dbheap</td>
<td>DB</td>
<td>mittel</td>
<td>32 - 524.288</td>
<td></td>
<td>Speicherseiten</td>
</tr>
<tr>
<td>pckcachesz</td>
<td>DB</td>
<td>hoch</td>
<td>-1; 32 - 128.000 (32 Bit);</td>
<td>-1</td>
<td>Speicherseiten</td>
</tr>
<tr>
<td>asheapsz</td>
<td>DBM</td>
<td>hoch</td>
<td>1 - 524.288</td>
<td>15</td>
<td>Speicherseiten</td>
</tr>
<tr>
<td>(locklist)</td>
<td>DB</td>
<td>hoch</td>
<td>4 - 524.288 (Unix)</td>
<td>100</td>
<td>Speicherseiten</td>
</tr>
<tr>
<td>(sheapthres_shr)</td>
<td>DB</td>
<td>hoch</td>
<td>250 - 2.097.152 (32 Bit),</td>
<td>20.000</td>
<td>Speicherseiten</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>250 - 2.147.483.647 (64 Bit)</td>
<td>20.000</td>
<td>Speicherseiten</td>
</tr>
</tbody>
</table>

Tabelle 5.4: Zusammensetzung des shared memory aus DB - Komponenten

inklusive geschätzten 10% Overhead. Für die im Rahmen dieser Diplomarbeit relevanten Betriebssysteme Linux (Entwicklungsumgebung) und AIX (Produktionsumgebung) stehen nach Abzug des Overheads insgesamt noch knapp 1,6 GB zur Verfügung. Ignoriert man aus Vereinfachungsgründen die in Tabelle 5.3 auf Seite 52 angegebenen Restriktionen und teilt jeden Pufferpool (Daten, Index, Temp) jeweils 500 MB (insgesamt also 1,5 GB) zu, so ist der noch maximal zulässige shared memory (1,6 GB, s. o.) quasi aufgebracht. Da bei dieser (im Sinne von maximalen Pufferpoolgrößen) bestmöglichen Konfiguration jeweils nur Bruchteile, d. h. wenige Prozent obiger drei Tabellenbereiche pufferbar sind, ist zu befürchten, dass viel Performanzpotenzial im Sinne einer Optimierung ungenutzt bleibt. Bei 76 (92) Byte pro Tupel sind rechnerisch ca. 6.580.000 (5.440.000) Tupel pufferbar, bei insgesamt 165.000.000 Tupeln sind das dann ca. 4% (3%).

Da 64 Bit Betriebssysteme (64 Bit Instanzen) keiner shared memory Restriktion unterliegen, sollte die DB nach Möglichkeit darauf aufsetzen, um vermutetes, brachliegendes Optimierungspotential besser auszunutzen.

Im konkreten Fall werden drei Pufferpools SYN_BP_DATA für die Daten, SYN_BP_IDX für die Indizes und SYN_DB_TMP für die temporären Daten mit jeweils 1 GB Kapazität angelegt. Bei Bedarf kann die Kapazität angepasst werden.

5.4 Multidimensionales Clustering (MDC)

Multidimensionales Clustering (MDC) zielt primär auf den Einsatz in data warehouses und großen Datenbankumgebungen ab sowie in OLTP - Umgebungen. Eine MDC - DBT ordnet Datensätze mit gleichen Dimensionsschlüsseln in Blöcke von zusammenhängenden Seiten an. D. h., Datensätze innerhalb eines Blöckes besitzen die gleichen Dimensionsschlüssel [BW03]. 'Gewöhnliche' Indizes (single dimension clustering [B]) können lediglich eindimensionaler clustern. Als Literaturverweise seien hier [BCHLMP03], [BCHMP03], [BCMP03] und [W03] genannt, von denen neben [BW03] einige Anregungen zur Dimensionsschlüsselbestimmung stammen (s. 5.4.1.1 auf Seite 54).

Die bildliche Darstellung der Duplikate anhand ihrer Koordinaten erfolgt durch die Webapplikation hier zweidimensional in Form einer Gerade, mit dem Koordinatenpaar <start-
KAPITEL 5. ÜBERLEGUNGEN ZUM PHYSISCHEM DATENBANKENTWURF

5.4.1 Richtlinien zur Dimensionsschlüsselbestimmung

Für die Auswahl und die Quantität der Dimensionsschlüssel gelten u. a. diese Empfehlungen:

1. Attribute, die in Bereichsabfragen oder IN-Listen-Klauseln vorkommen sind potenzielle Kandidaten.
2. Group by / order by-Klausel-Attribute sind potentielle Kandidaten.
3. Fremdschlüsselattribute sind potentielle Kandidaten.
4. In Abfragen oft zusammen auftretende Attribute sind potentielle Kandidaten und zu einer Dimension zusammenzufassen.
5. Unique-Attribute / Primärschlüssel sind keine guten Kandidaten.

Trotz Berücksichtigung dieser Richtlinien ist die Suche nach 'den' Kandidaten oft ein trial and error-Verfahren. Bei falscher Kandidatenauswahl besteht zudem die latente Gefahr, dass aufgrund ungeeigneter Attribute die Seiten nur unzureichend gefüllt werden, was bei großen Datenmengen schnell zu enormer Speicherplatzverschwendung führt: Eine schlechte Kandidatenauswahl mit halbleeren Seiten verdoppelt den hier veranschlagten Speicherplatzbedarf von ca. 17,4 GB auf ca. 35 GB.

5.4.1.1 Bestimmung der Dimensionsschlüssel


54
KAPITEL 5. ÜBERLEGUNGEN ZUM PHYSISCHEN DATENBANKENTWURF


In einer MDC - DBT bildet jede eindeutige Kombination von Dimensionsschlüsselwerten eine logische Zelle, die sich physisch aus Blöcken von Seiten zusammensetzt und ein Block eine Menge aufeinander folgender Seiten auf dem Datenträger ist.

5.4.1.2 Beschreibung der Testumgebung


5.4.1.3 Testablauf

Nach dem Laden der Datensätze in beide DBT enthielten diese jeweils 2259751 (6587646) Tupel mit der param_id 3 (4), insgesamt also jeweils knapp 9 Mio. Tupel. Unter Verwendung von

\[ \text{db2advis -d SYNTENY -s "<query>"} \]

2Man beachte, dass die je zwei MDC - Attribute nicht zu einer Dimension zusammengefasst wurden
wurde an charakteristischen, d. h. oft verwendeten Abfragen die Zugriffs kosten (timerons) und ggf. zugriffsbeschleunigende Indizes bestimmt (s. [IBM VI]). Für zwei Abfragen, 5.1 und 5.2 wird dies kurz skizziert:

**Abfrage 5.1**  
`SELECT carrier_id, param_id, count(*) FROM duplicates.<tablename> GROUP BY carrier_id, param_id`

**Abfrage 5.2**  
`SELECT * FROM duplicates.<tablename> WHERE carrier_id = 23 AND param_id = 3 AND startpos1 >= 20000000 AND endpos1 <= 40000000 AND startpos2 >= 20000000 AND endpos2 <= 40000000`

Die Ergebnisse sind in den Tabellen 5.5 und 5.6 angegeben.

<table>
<thead>
<tr>
<th>Abfrage</th>
<th>timeron ohne Indizes</th>
<th>timeron mit aktueller Lösung</th>
<th>Verbesserung</th>
<th>Liste empfohlener Indizes</th>
</tr>
</thead>
<tbody>
<tr>
<td>mdc - DBT</td>
<td>258,3903</td>
<td>258,3903</td>
<td>0,00%</td>
<td>∅</td>
</tr>
<tr>
<td>nicht - mdc - DBT</td>
<td>579741,7500</td>
<td>579741,7500</td>
<td>0,00%</td>
<td>∅</td>
</tr>
</tbody>
</table>

**Tabelle 5.5: db2advis - Ergebnisse für Abfrage 5.1**

Für Abfrage 5.1 ist der bisherige Datenzugriff bereits optimal. Beachtlich ist der time-ron-Unterschied um den Faktor 2244. Die prozentuale Angabe der Verbesserung ist nach Auffassung des Autors eher intuitiv irreführend: Die rund 90% Verbesserung in Tabelle 5.6 erwecken den Eindruck einer Verbesserung fast um Faktor 2, obwohl es sich fast eine Verbesserung um Faktor 10 handelt (entspricht rund 900% Verbesserung).

<table>
<thead>
<tr>
<th>Abfrage</th>
<th>timeron ohne Indizes</th>
<th>timeron mit aktueller Lösung</th>
<th>Verbesserung</th>
<th>Liste empfohlener Indizes</th>
</tr>
</thead>
<tbody>
<tr>
<td>mdc - DBT</td>
<td>252,3986</td>
<td>26,6320</td>
<td>89,45%</td>
<td>s. 5.2</td>
</tr>
<tr>
<td>nicht - mdc - DBT</td>
<td>561884,8750</td>
<td>2988,3706</td>
<td>99,47%</td>
<td>s. 5.1</td>
</tr>
</tbody>
</table>

**Tabelle 5.6: db2advis - Ergebnisse für Abfrage 5.2**

Die Zugriffsverbesserung durch das Anlegen der empfohlenen Indizes

**Index 5.1** `PARAM_ID ASC, CARRIER_ID ASC, STARTPOS2 ASC, STARTPOS1 ASC, ENDPOS2 ASC, ENDPOS1 ASC ALLOW REVERSE SCANS`

**Index 5.2** `PARAM_ID ASC, CARRIER_ID ASC, STARTPOS2 ASC, STARTPOS1 ASC, ENDPOS2 ASC, ENDPOS1 ASC, REDUNDANCE_ID ASC, NATIVE_CLUSTER_ID ASC, PRCNT_IDENT ASC, LENGTH ASC, EVALUTE ASC, MISMATCHES ASC, PAL_ID ASC ALLOW REVERSE SCANS`

wurde für Abfrage 5.2 mit Hilfe von db2advis erneut überprüft. Wegen Speicherplatzmangels musste der Index 5.2 auf den Index 5.1 reduziert werden. Warum der Index 5.2 sich derart vom Index 5.1 unterscheidet ist nicht nachvollziehbar. Die nunmehrigen Ergebnisse sind in Tabelle 5.7 auf Seite 57 angegeben:

Der modifizierte Index ist auch optimal (keine Verbesserung möglich). Die Ergebnisse für Abfrage 5.2 und deren Modifikation sind quasi ein Spiegelbild von den Ergebnissen der Abfrage 5.1. Die MDC - timeron - Werte sind im Vergleich zu dem nicht - MDC - timeron
KAPITEL 5. ÜBERLEGUNGEN ZUM PHYSISCHEN DATENBANKENTWURF

Abfrage 5.2/5.2

<table>
<thead>
<tr>
<th>mdc - DBT</th>
<th>timeron ohne Indizes</th>
<th>timeron mit aktueller Lösung</th>
<th>Verbeserung</th>
<th>Liste empfohlener Indizes</th>
</tr>
</thead>
<tbody>
<tr>
<td>30075,2871</td>
<td>30075,2871</td>
<td>0,00%</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>562501,8125</td>
<td>562501,8125</td>
<td>0,00%</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Tabelle 5.7: db2advis - Ergebnisse nach Erzeugung der empfohlenen Indizes für Abfrage 5.2

- Wert deutlich besser.

Frappierend sind allerdings die timeron - Ergebnisse nach Anlegen der Indizes mit anschließendem runstats: Während die timeron - Werte für die non - MDC - DBT stagnieren, verschlechtern sich die der MDC - DBT drastisch (und bieten reichlich Grund zu Spekulationen über die Ursache).

Durch das Anlegen zahlreicher Indizes (mit anschließender runstats - Ausführung) unter Zuhilfenahme von db2advis bleibt festzuhalten: Bei bereits angelegten Indizes, die viele (Index-)Attribute mit noch nicht angelegten und deshalb ähnlichen Indizes gemein haben, profitieren letztere durch die bereits zuletzt angelegten Indizes. Der aufgeworfenen Frage, ob db2advis für eine Menge von Abfragen einen einzelnen, aber für jede Abfrage optimalen Index erstellen kann, soll nicht nachgegangen werden.

Die für jede Abfrage unter explain all with snapshot for "<query>" zu erstellenden Zugriffspläne waren nicht verfügbar. Die während des Aufbaus der Tests an einem anderen Rechner erstellten Zugriffspläne zeigen, dass neben Indexscans vor allem Tabellenscans (Abfrage 5.1) von sinnvoll angelegten MDC - DBT profitieren. In Abschnitt 8.6 auf Seite 103 werden Indizes und Zugriffspläne noch einmal behandelt.

5.4.2 Konfiguration des SYN_TS_DATA - Tabellenbereichs

Neben der performanzbeeinflussenden Festlegung der Dimensionsschlüssel, ist die Festlegung des speicherplatzbeeinflussenden blocking factors - der dem extent size entspricht - und u. U. auch performanzbeeinflussend [IBM]. Der Standardwert von 32 ist für die Duplikatdaten mit obigen Dimensionsschlüsseln ebenfalls ungeeignet: So verbrauchten nach einem erfolgreichen LOAD allein die rund 80 Mio. Repeats von homo sapiens und pan troglodytes bei einem (für nicht - MDC - DBT) 'empfohlenen' extent size von 64 annähernd 4 Mio Speicherseiten, die nach der Kalkulation (Abschnitt 5.1.1 auf Seite 49) beinahe für alle Duplikate ausreichen sollten und mindestens 15 GB Speicherplatzverschwendung zur Folge hat!

Zwar ist der extent size - Parameterwert prinzipiell veränderbar, aber nicht nachträglich über alter tablespsace, sodass der geeignetste Parameterwert bereits bei der Erstellung (create regular tablespsace) angegeben werden sollte. Als Vorlage für die Festlegung eines geeigneteren extent size - Parameterwertes diente [GM02].

5.4.2.1 Beschreibung der Testumgebung

Für den Vergleich wurden zwei Tabellenbereiche SYN_TS_DATA_2 und SYN_TS_DATA_32 (s. u.) auf demselben Datenträger angelegt, die sich aus jeweils zwei Containern a 1 GB zusammensetzen. Die Zahlenwerte entsprechen dem verwendeten extent size und letzterer ist DB2 - Voreinstellung. Im erstgenannten (zweitgenannten) Tabellenbereich wurde eine einzelne DBT duplicates.palindromes_2 (duplicates.palindromes_32) erstellt, natürlich ohne Fremdschlüsselbeziehungen und den Dimensionsschlüsseln ((carrier_id, param_id), (mdc_startpos1, mdc_endpos1), (mdc_startpos2, mdc_endpos2)).
5.4.2.2 Testablauf

Als Testdaten wurden die 8.852.773 Palindromdatensätze vom humanen X-Chromosom und vom Y-Chromosom der Spezies pan troglodytes in beide DBT geladen.

Die Tabelle 5.8 gibt einige über list tablespaces show detail gewonnene Informationen zu beiden Tabellenbereichen vor der Dateneintragung wieder.

<table>
<thead>
<tr>
<th></th>
<th>SYN_TS_DATA_2</th>
<th>SYN_TS_DATA_32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seiten insgesamt</td>
<td>524288</td>
<td>524288</td>
</tr>
<tr>
<td>verwendete Seiten</td>
<td>14</td>
<td>224</td>
</tr>
<tr>
<td>Speichergröße (Byte)</td>
<td>4096</td>
<td>4096</td>
</tr>
<tr>
<td>extent size (Byte)</td>
<td>2</td>
<td>32</td>
</tr>
<tr>
<td>Anzahl Container</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Tabelle 5.8: Informationen zu den Tabellenbereichen vor dem Laden

Die Angaben unter verwendete Seiten sind dabei schon aufschlussreich: Man erkennt, dass 224 / 14 = 16 = 32 / 2 gilt und die verwendete Seiten offensichtlich (auch) vom extent size - Parameterwert abhängen.

Die Tabelle 5.9 enthält die Ergebnisse nach der Eintragung der knapp 9 Mio. Datensätze. Die 276128 - 254446 = 21682 zusätzlichen Speicherseiten entsprechen einem Mehrbedarf von rund 8,6%.

<table>
<thead>
<tr>
<th></th>
<th>SYN_TS_DATA_2</th>
<th>SYN_TS_DATA_32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seiten insgesamt</td>
<td>524288</td>
<td>524288</td>
</tr>
<tr>
<td>verwendete Seiten</td>
<td>254446</td>
<td>276128</td>
</tr>
<tr>
<td>Seitengröße (Byte)</td>
<td>4096</td>
<td>4096</td>
</tr>
<tr>
<td>extent size (Byte)</td>
<td>2</td>
<td>32</td>
</tr>
<tr>
<td>Anzahl Container</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

Tabelle 5.9: Informationen zu den Tabellenbereichen vor dem Laden

Die vorherige Wahl des extent size Parameters von 64, der in Abhängigkeit dieses Datenumfangs für nicht-MDC-DBT empfohlen wird, beansprucht in Bezug auf die in Abschnitt 5.1.1 auf Seite 49 durchgeführten Kalkulationen sogar den doppelten Speicherbedarf.

Das Testergebnis ist wegen einer in etwa gleichen Anzahl von Repeats und Palindromen pro Chromosom auf die Repeats übertragbar. Ein kleiner extent size - Parameterwert ist daher empfehlenswert.

5.4.3 Bestimmung der Seitengröße


\[
\text{Byte} \frac{\text{Datensatz}}{92} \cdot 255 \text{ Datensätze} = 23460 \text{ Byte.}
\]

D.h., bei Verwendung von 32 KB-Seiten bleiben mindestens 32768 Byte - 23460 Byte ≈ 9300 Byte ungenutzt. 32 KB-Seiten sind folglich ungeeignet. Für die in Frage kommenden restlichen drei ist zu berücksichtigen, dass u. U. deutlich weniger Datensätze pro Seite

58
eingetragen werden. Dem Vorteil eines schnelleren Datenzugriffs bei größeren Seite steht der
gravierendere Nachteil einer Speicherplatzverschwendung gegenüber. Da diese Entscheidung
letztendlich von den konkreten Daten abhängt, wird hier auf 4 KB - Seiten und damit kleins-
te Einheit zurückgegriffen. Bei 4028 nutzbaren Bytes pro 4 KB - Seiten können demzufolge
maximal
\[
\left\lfloor \frac{4028 \text{ Byte}}{92 \text{ Byte/Datensatz}} \right\rfloor = 43 \text{ Datensätze}
\]
eingetragen werden.

### 5.5 Dateneinspeisung

Erst eine Höhersetzung des `util_heap_sz` bei der DB - Konfiguration ermöglichte die Ausnut-
zung der vollen LOAD - Parallelität. Wie in der Tabelle 5.10 angegeben, vergehen mehrere
Stunden bis zur vollständigen Einstellung der Daten. Zur Zeitmessung wurde wie bisher
`/usr/bin/timex` benutzt.

<table>
<thead>
<tr>
<th></th>
<th>real (s)</th>
<th>user (s)</th>
<th>sys (s)</th>
<th>%usr</th>
<th>%sys</th>
<th>%wio</th>
<th>%idle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repeats</td>
<td>17.330,96</td>
<td>0,79</td>
<td>2,40</td>
<td>4</td>
<td>2</td>
<td>38</td>
<td>55</td>
</tr>
<tr>
<td>Palindrome</td>
<td>79.958,37</td>
<td>0,82</td>
<td>2,57</td>
<td>1</td>
<td>2</td>
<td>65</td>
<td>32</td>
</tr>
</tbody>
</table>

Tabelle 5.10: Laufzeiten für `load` und `set integrity`

Die Zeitdiskrepanz zwischen Repeats (knapp 5 Stunden) und Palindromen (gut 22 Stun-
den) ist - wenn überhaupt - mit einer unterschiedlichen Auslastung des Lehrstuhlrechners
(Produktionsumgebung) zu erklären. Im Vergleich zum handelsüblichen, eher dürftig aus-
gestatteten privaten Standalone - Rechner (Entwicklungsumgebung) ist letzterer aufgrund
einer wesentlich besseren CPU - Auslastung in Bezug auf Performanzkriterien durchaus
konkurrenzfähig. Es spricht daher einiges dafür, dass eine für die Ziele dieser Diplomarbeit
abgestellte, gut ausgestattete Workstation ("SYNTENY - Server") unter Performanzaspek-
ten wohl eine optimale Wahl ist.

Nach Möglichkeit ist der Server in Eigenregie mit Systemadministratorrechten zu betreiben.
Dies führt zur Auseinandersetzung mit der vorhandenen Hardware und hilft auf der einen
Seite, richtige Entscheidungen beim DB - Design zu treffen. Auf der anderen Seite beginnt
die DB - Optimierung bereits durch entsprechende Konfiguration des Betriebssystems.

Das Laden der unmaskierten Repeat- und Palindromdatensätze erfolgte chromosomenweise
(dateiweise) mit anschließender `set integrity` - Anweisung über ein Skript. Wegen erhöhtem
Speicherplatzbedarf - oft zu Lasten der Protokollierung - kann das Laden bei Problemen dann
chromosomenweise und damit dateiweise fortgesetzt werden. Insbesondere in der Entwick-
 lungsphase wurden mehrere LOAD - Fehlschlüsse mit jeweils seitenspezifischen Fehlereinträgungen
in `db2diag.log` verzeichnet und konnte letztendlich nur durch ein Neuanlegen der DB SYN-
 TENY behoben werden.

Es hat den Anschein, dass insbesondere `generated` - Attribute die LOAD - Performance beein-
trächtigen, zumal darüber hinaus bspw. für die Berechnung von `prent_ident` mehrere Daten-
typkonvertierungen erfolgen. Die Überprüfung von Fremdschluesselbeziehungen oder diversen
Restriktionen scheint sich kaum auf die Laufzeit auszuwirken, allerdings sind `unique con-
straints` wegen des Datenumfangs wohl besser zu vermeiden.

Das Eintragen aller 73 DNA - Sequenzen scheiterte an den zu kleinem temporären Tabellen-
bereich `SYN_TS_TEMP`. Mit speziesspezifischen Sequenzimport liess sich das Problem mit relativ
kurzen Laufzeiten (s. Tabelle 5.11) lösen.

<table>
<thead>
<tr>
<th>Chr.</th>
<th>real (s)</th>
<th>user (s)</th>
<th>sys (s)</th>
<th>%usr</th>
<th>%sys</th>
<th>%wio</th>
<th>%idle</th>
</tr>
</thead>
<tbody>
<tr>
<td>homo sapiens (unmasked)</td>
<td>24</td>
<td>652,46</td>
<td>0,02</td>
<td>0,09</td>
<td>0</td>
<td>3</td>
<td>14</td>
</tr>
<tr>
<td>homo sapiens (masked)</td>
<td>24</td>
<td>751,86</td>
<td>0,01</td>
<td>0,09</td>
<td>0</td>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td>pan troglodytes (unmasked)</td>
<td>25</td>
<td>718,16</td>
<td>0,02</td>
<td>0,08</td>
<td>0</td>
<td>3</td>
<td>14</td>
</tr>
</tbody>
</table>

Tabelle 5.11: Laufzeiten für speziesweisen Sequenzimport

Das Füllen der übrigen DBT ist unproblematisch, aber wegen vieler semantischer Abhängigkeiten (Fremdschlüsselbeziehungen, ...) und entsprechender Datencodierung über Identifer kaum automatisierbar.

Mittels runstats, reorgchk und reorg wurden einige Datenbankobjekte überprüft und ggf. aktualisiert sowie durch autoconfigure die DB2-Konfiguration an den vorhandenen Datenumfang angepasst.

5.6 Kapitelzusammenfassung und Fazit


Es hat sich gezeigt, dass für die Bewältigung der Datenmasse erhebliche Vorkehrungen und Grundsatzentscheidungen zu treffen sind, die hier aber nur in groben Zügen dargelegt sind.
Kapitel 6

Datenkontrolle & -aufbereitung

Dieses Kapitel behandelt die Datenkontrolle, Datenaufbereitung und Datenauswertung der DNA-Sequenzen (Abschnitt 6.1) und Duplikatergebnisse (Abschnitt 6.2 auf Seite 63).

6.1 DNA-Sequenzen


Datenkontrolle: Kontrolle auf IUPAC-Basen und Fremdzeichen

Datenaufbereitung: Positionsbestimmung von unbekannten Sequenzabschnitten

Datenauswertung: statistische Auswertung unbekannter Sequenzabschnitte

Abbildung 6.1: Schema der Verarbeitungskette der DNA-Sequenzen

6.1.1 Datenkontrolle und -aufbereitung

Nach Definition 2.4 auf Seite 18 ist \( S_N \) (S) eine (lückenlose) DNA-Sequenz gdw. \( S_N \in B_N^+ \) (\( S \in B^+ \)). Die IUPAC-Konvention [CB85] stellt aber neben ‘A’, ‘C’, ‘G’, ‘T’ und dem universellsten Platzhalter ‘N’ noch spezifischere Platzhalter für unbestimmte DNA-Sequenzabschnitte zur Verfügung (s. Tabelle 6.1 auf Seite 62).

Normalerweise ist die Annahme berechtigt, dass für jede der 73 DNA-Sequenzen ein \( S_N \in B_N^+ \) bzw. \( S \in B^+ \) existiert. Ein Auftreten anderer Basen alias Platzhalter erschwert Sequenzvergleiche: Für \( S_N \) oder \( S \) beschränkt sich der Sequenzvergleich auf die Feststellung der (Un-)Gleichheit zweier Basen; bei Vorliegen von weiteren Basen nach der IUPAC-Konvention ist ein solcher Vergleich unzureichend und bedarf anderer Vergleichsstrategien.

Beinhalten die chromosomalen DNA-Sequenzen in geringer Anzahl andere Platzhalter als
### Tabelle 6.1: IUPAC - Konvention zur Codierung von Basen

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Bedeutung</th>
<th>Symbol</th>
<th>Bedeutung</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A (Adenin)</td>
<td>N</td>
<td>A oder C oder G oder T</td>
</tr>
<tr>
<td>B</td>
<td>C oder G oder T; nicht A</td>
<td>R</td>
<td>A oder G</td>
</tr>
<tr>
<td>C</td>
<td>C (Cytosin)</td>
<td>S</td>
<td>C oder G</td>
</tr>
<tr>
<td>D</td>
<td>A oder G oder T; nicht C</td>
<td>T</td>
<td>T (Thymin in DNA), (Uracil in RNA)</td>
</tr>
<tr>
<td>G</td>
<td>G (Guanin)</td>
<td>V</td>
<td>A oder C oder G; nicht T</td>
</tr>
<tr>
<td>H</td>
<td>A oder C oder T; nicht G</td>
<td>W</td>
<td>A oder T</td>
</tr>
<tr>
<td>K</td>
<td>G oder T</td>
<td>Y</td>
<td>C oder T</td>
</tr>
<tr>
<td>M</td>
<td>A oder C</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

N, so werden diese durch 'N' substituiert. Bei Aufspürung von Fremdzeichen wäre zu entscheiden, wie dann mit den DNA- Sequenzen verfahren wird. Wie in Abschnitt 4.2.2.1 auf Seite 10 geschildert, sind die Bordmittel von DB2 zur Aufdeckung und Positionsbestimmung von Platzhaltern oder Fremdzeichen unzureichend. Deswegen musste auf eine externe Applikation in Form eines C-Programmes zurückgegriffen werden. Dieses Programm get\_dna\_gaps erwartet die DNA-Sequenzen in einer einzeiligen Datei <dna\_seq\_file> und wird über

```
get\_dna\_gaps <carrier\_id> <dna\_seq\_file> <results\_file>
```

#### 6.1.2 Datenauswertung

Die Ergebnisse, die in den Tabellen C.1 auf Seite 129, C.2 auf Seite 130 und C.3 auf Seite 130 angegeben sind, lassen sich wie folgt zusammenfassen:

- Alle vorliegenden chromosomalen DNA-Sequenzen beinhalten keine Fremdzeichen.
- Die Anzahl der unbestimmten Sequenzabschnitte (Gaps) der unmaskierten DNA-Sequenzen von *homo sapiens* liegt chromosomenweise bei maximal einigen Dutzend. Die prozentuale, kumulierte Gaplänge pro Chromosom variiert sehr stark und liegt überwiegend (deutlich) im einstelligen Bereich. Ausschließlich bei Chromosom drei wurde an den Positionen 60.787.869 (‘M’) sowie 60.788.098 und 60.788.099 (jeweils ‘R’) eine Notation im IUPAC-Format festgestellt, die jeweils eine Substitution durch ‘N’ nach sich zog.
- Bei den maskierten DNA-Sequenzen von *homo sapiens* konnten obige Notationen nicht wiedergefunden werden, weil diese Positionen in einem maskierten Bereich liegen. Verglichen mit den obigen Ergebnissen liegen die Anzahl der Gaps (inklusive maskierter Bereiche) pro Chromosom jeweils um Größenordnungen darüber, was sich zwangsläufig in der prozentualen, kumulierten Gaplänge widerspiegelt: Nunmehr liegt diese - von wenigen Ausnahmen abgesehen - jeweils bei ca. 50%. In Abb. 6.2 auf Seite 63 geben die Balken den prozentualen

---

1 gemeint sind alle IUPAC - Basen außer ’A’, ’C’, ’G’, ’T’, ’N’

Abbildung 6.2: Sequenzstatistik

Wegen der starken Korrelation in Bezug auf die Chromosomenanzahl als auch der DNA-Gesamtlängen von *pan troglodytes* mit *homo sapiens* ist ein interspezifischer Vergleich durchaus aufschlussreich: Die Anzahl der Gaps liegt einerseits deutlich über (unter) denen der unmaskierten (maskierten) DNA-Sequenzen von *homo sapiens*, andererseits trifft dies für die kumulierten, prozentualen Gaplängen nur großteils zu, denn punktuell sind kaum noch Unterschiede auszumachen. Es liegt daher auf der Hand, dass demzufolge auch die Anzahl der Duplikate zwischen den Werten der korrespondierenden Chromosomen von *homo sapiens* liegen.

Anhand der einzelnen kumulierten, prozentualen Gaplängen erklären sich auch die signifikanten Unterschiede bezüglich der Quantität der Duplikate: Bei tendenziell sensitiver VMATCH-Parameterwahl erhält man bei geringer kumulierter Gaplänge überproportional viele Duplikate.


### 6.2 Duplikatergebnisse

In Abb. 6.3 auf Seite 64 sind die einzelnen Verarbeitungsschritte für die Duplikatergebnisse angegeben. Abgesehen von der Datenkontrolle tangieren alle weiteren Verarbeitungsschritte nur Repeats und Palindrome. Die besonderen Zusatzigenschaften von Tandems (keine

Datenkontrolle: Überprüfung der Sequenzen der Duplikatarme auf Korrektheit (6.2.1) 
Datenaufbereitung: Redundanzerkennung und -beseitigung bei Repeats und Pal. (6.2.2.1) 
Datenaufbereitung: Zusammenfassen sich teilw. überlappende Repeats bzw. Pal. (6.2.2.2) 
Datenaufbereitung: Neubestimmung der Anz. Mismatches dieser Repeats und Pal. (6.2.2.3) 
Datenaufbereitung: erweitertes Zusammenfassen von Repeats bzw. Pal. (6.2.2.4) 
Datenaufbereitung: Kennzeichnung nichtred. (Sub-)Repeats bzw. (Sub-)Pal. (6.2.2.5)

Abbildung 6.3: Schema der Verarbeitungskette von Repeats und Palindromen

Das detaillierte Verständnis der obigen Verarbeitungsschritte ist für die folgenden Kapitel ebenfalls nicht notwendig. Mit Hilfe der folgenden Zusammenfassung lässt sich das restliche Kapitel überspringen:

**Redundanzerkennung und -beseitigung bei Repeats und Palindromen**


**Zusammenfassen sich teilweise überlappende Repeats bzw. Palindrome (native clustering)**

VMATCH erlaubt bei Repeats und Palindromen maximal 10 Mismatches. Diese Restriktion beruht offensichtlich auf Laufzeitgründen, nicht aber auf biologischen Sachverhalten. D. h., falls der Präfix der DNA-Sequenzen beider Repeatarme eines Repeats jeweils auch Suffix der DNA-Sequenzen beider Repeatarme eines anderen Repeats ist, so lassen sich beide Repeats zu einem Repeat vereinen. Dieses Vereinen wird hier als native clustering bezeichnet und ist ebenso auf Palindrome anwendbar.

**Neubestimmung der Anz. Mismatches dieser Repeats und Palindrome**

Das Attribut mismatches vermerkt für jedes Repeat und Palindrom die Anzahl der Basenabweichungen beider Duplikatarme. Weil sich aber die genauen Mismatchpositionen daraus nicht ableiten lassen, lässt sich auch nicht feststellen, wie viele Mismatches sich überlappende Repeats und Palindrome teilen (wenn überhaupt). Die Neubestimmung vergleicht die Duplikatarmsequenzen basenweise, summiert Basenungleichheiten auf und trägt diese dann ein.
Erweitertes Zusammenfassen von Repeats bzw. Palindromen

*(extended native clustering)*

Nicht nur sich teilweise überlappende Repeats oder Palindrome lassen sich clustern, sondern prinzipiell auch Repeats und Palindrome mit einem benachbarten Repeat bzw. Palindrom. In Abhängigkeit eines noch festzulegenden Kriteriums für erweitertes Zusammenfassen nennt sich dieser Prozess *extended native clustering*.

Kennzeichnung von nichtred. (Sub-)Repeats bzw. (Sub-)Palindromen

(Sub-)Repeats und (Sub-)Palindrome aus diversen VMATCH - Läufen mit unterschiedlicher Parametervariation (s. Tabellen 4.6 auf Seite 37 und 4.7 auf Seite 38) sind a priori nicht redundant, auch falls ein Repeat (Palindrom) mit stringenterer Parameterwahl durchaus vollständig in einem längeren Repeat (Palindrom) enthalten ist. Da bei einer grafischen Ausgabe (Vgl. Abb. 6.6 auf Seite 67) beider Repeats (Palindrome) wegen der Überlagerung nur das längere erkennbar ist, lassen sich (Sub-)Repeats ((Sub-)Palindrome) dann über eine entsprechende Kennzeichnung schon bei der Abfrage herausfiltern.

6.2.1 Datenkontrolle


Eine systematische Untersuchung aller Duplikatarmsequenzen bestätigt entweder die Korrektheit (fast) aller Duplikatergebnisse und lenkt die Ursachenerkundung auf biologisches Terrain oder lässt auf einen methodischen Fehler schließen. Weil die DB SYNTENY sowohl die DNA - Sequenzen als auch die Duplikatarmkoordinaten speichert, lässt sich mit einigen Funktionserweiterungen die Sequenzkontrolle in Form einer SQL PL Prozedur unterbringen.

Der ersten Funktion, `sequence.dna_count_mismatches`, werden als Parameter die beiden Duplikatarmsequenzen übergeben sowie als Rückgabewert die Anzahl der gefundenen Mismatches (Mispairs). Stimmt der Rückgabewert mit dem vermerkten Wert des Attributs `mismatches` dieses Tupels nicht überein, so wird der Attributwert des entsprechenden Primärschlüssels (`repeat_id`, `palindrome_id`, ...) in einer gesonderten DBT eingetragen.

Die zweite Funktion, `sequence.dna_revcom`, findet ‘nur’ bei den Palindromen Verwendung: Hier muss ausgehend von den DNA - Sequenzen erst das reverse complement genau eines Palindromarmes bestimmt werden, ehe dann beide Palindromarmsequenzen an `sequence.dna_count_mismatches` übergeben werden. Vor allem aus Performanzgründen wurden beide Funktionen in C implementiert und als SQL - Funktion zur Verfügung gestellt. Mittels der SQL PL Prozeduren

2Bei Tandems und supermaximalen Repeats fehlt dieses Attribut, da diese Duplikate 0 Mismatches aufweisen müssen

3und auf der AIX - Produktionsumgebung mittels `xlc -q64 64bittig` compiliert
Abbildung 6.4: Unterschiedliche lange Palindromsequenzen mit gleichen Startpositionen

sequence.check_duplicatetype._seq(<carrier_id>, <param_id>)

... erfolgt die Kapselung und der Aufruf der Sequenzkontrolle.

Der durch die C - Implementation gewonnene Performanzschub manifestiert sich in den sehr guten Laufzeiten samt Prozessorauslastung, die für Repeats und Palindrome in Tabelle B.9 auf Seite 128 und B.10 auf Seite 128 angegeben sind. Überraschend weist die zusätzliche reverse - complement - Bestimmung der Palindrome bei vergleichbarer Duplikatanzahl keine Auswirkungen auf die Laufzeit auf.

Die entscheidende Erkenntnis dieser Datenkontrolle ist die Korrektheit aller (!) Duplikate. Demnach findet sich die Erklärung zu Abb. 6.4 im biologischen Kontext: Zum einen im hochrepetitiven Charakter (CCTT, CTTT) und den entgegengesetzten Leserichtungen von Palindromarmen.

6.2.2 Datenaufbereitung

6.2.2.1 Redundanzerkennung und -beseitigung

Für die Bestimmung redundanter Duplikate reicht die Feststellung allein nicht aus, dass ein kürzeres Duplikat in einem längeren Duplikat vorkommt. In Abb. 6.5 auf Seite 67 sind die beiden kürzeren Repeatarme TC (U ∈ R) jeweils in den längeren Repeatarmen ACGTGC-TAGC und ACGTCTCAACG (V ∈ R) vollständig enthalten und redundant. Der Präfix ACG eines jeden Repeatarmes von V ist zugleich Suffix beider Repeatarme und folglich ebenfalls ein (Sub-)Repeat, W ∈ R, jedoch ist es nicht redundant. Diese Erkenntnisse erschließen sich bei entsprechender Betrachtung von Abb. 6.6 auf Seite 67. Hier sind die einzelnen Alignierungen mit Kennzeichnung der Leserichtung als Pfeile eingezeichnet und die zwei Mismatches als Kleinbuchstaben symbolisiert. Redundante Duplikate sind als überlagerte(r) Pfeil(e) identifizierbar. Offensichtlich ist W auch ein (Sub-)Repeat, aber ein nicht-r-redundantes, da es nicht überlagert wird.

Darüber hinaus ist an W eine achsensymmetrische 'Doppelung' ersichtlich: Beide Repeatpfeile von W spiegeln sich an der Winkelhalbierenden, die hier (zufällig auch) durch V
KAPITEL 6. DATENKONTROLLE & -AUFBEREITUNG

Abbildung 6.5: Repeat (V) mit (Sub-) Repeats (U, W)

repräsentiert wird. D. h., dass man für gewöhnlich die Duplikate in der oberen 'Dreiecksmatrix' einzeichnet. Diese Symmetrieigenschaft gilt auch für Tandems und supermaximal Repeats und mit leichten Modifikationen auch für Palindrome (s. u.).

Durch die DB - Restriktion der Attributwerte $\text{startpos}_1 < \text{startpos}_2$ wird ohne weitere Vergleiche und Performanzeinbußen das Duplikat in der oberen Dreiecksmatrix abgebildet.

Abbildung 6.6: 2D - Darstellung von Abb. 6.5

67
Die optische Identifikation von redundanten Duplikaten gelingt durch das Auffinden überlagerter Pfeile. Doch wie lässt sich das auf mathematischem Weg bewerkstelligen? Über den spezifischen Diagonal- und Attributwert $\text{dia}_\text{value}$ der Repeats und Palindrome.

Wie aus Abb. 6.5 zu entnehmen ist, setzen sich $U$, $V$ und $W$ aus den Repeatarmen $U_1[m+5\ldots m+6] \in B^2$, $U_2[n+5\ldots n+6] \in B^2$, $V_1[m\ldots m+10] \in B^{11}$, $V_2[n\ldots n+10] \in B^{11}$ sowie $W_1[m\ldots m+2] \in B^3$ und $W_2[n+8\ldots n+10] \in B^3$ zusammen. Als korrespondierende zweidimensionale Koordinaten ergeben sich nach dem Muster $(\text{startpos}_1, \text{endpos}_1)$, $(\text{endpos}_2, \text{startpos}_2)$):

$U((m+5,n+5),(m+6,n+6))$, $V((m,n),(m+10,n+10))$ und $W((m,n+8),(m+2,n+10))$.

Wegen der gegenläufigen Leserichtung von Palindromarmen erfolgt die grafische Darstellung als Antidiagonale, nach dem Koordinatenschema $(\text{startpos}_1, \text{endpos}_2)$, $(\text{endpos}_1, \text{startpos}_2)$).

**Definition 6.1 (spezifische Diagonalwerte $\Delta, \delta$ (dia\_value))**  Sei $R \in \mathbb{R}$ mit den Repeatarmen $R_1[m\ldots m+l] \in B^{l+1}$ und $R_2[n\ldots n+l] \in B^{l+1}$ mit $m \in \mathbb{N}^+$, $n \in \mathbb{N}^+$ und o. B. d. A. $m < n$. Dann ist der für Repeats spezifische Diagonalwert $\Delta_R$ definiert als

$$\Delta_R := n - m$$  \hspace{1cm} (6.1)

und $\Delta_R \in \mathbb{N}^+$.

Sei $P \in \mathbb{R}$ mit den Palindromarmen $P_1[i\ldots i+l] \in B^{l+1}$ und $P_2[j\ldots l+l] \in B^{l+1}$ mit $i \in \mathbb{N}^+$, $j \in \mathbb{N}^+$ und o. B. d. A. $i < j$. Dann ist der für Palindrome spezifische Diagonalwert $\delta_P$ definiert als

$$\delta_P := i + (j + k).$$  \hspace{1cm} (6.2)

Als spezifische Diagonalwerte $\Delta_U, \Delta_V, \Delta_W$ erhält man nach Formel (6.1):

$$\Delta_U = n + 5 - (m + 5)$$

$$= n - m$$

$$\Delta_V = n - m$$

$$\Delta_W = n + 8 - m$$

$$= (n - m) + 8$$

Weil $\Delta_U = \Delta_V$ und $m \leq m+5 \leq m+6 \leq m+10$ ist U redundant. Wegen $\Delta_V \neq \Delta_W$ ist W nicht redundant. Der Betrag der Differenz

$$|\Delta_V - \Delta_W| = |(n - m) - (n - m + 8)|$$

$$= |n - m - n + m - 8|$$

$$= 8$$

ist die sogenannte Manhattandistanz (Vgl. [A03]).

Ist die Definition von $\Delta$ noch nachvollziehbar, so ist die Definition von $\delta$ vergleichsweise weniger plausibel. Klarheit sollte jedoch die Betrachtung des eingerahmten Palindroms $P$ mit den Palindromarmen $P_1[m\ldots m+11] = \text{AACGTGTCTAGC} \in B^{12}$, $P_2[n,n+11] = \text{CAGTGCATGTTC}$.
Abbildung 6.7: (Eingerahmtes) Palindrom mit eingetragenen Diagonalwerten

GCTTGAGACGGTT \( \in \mathbb{B}^{12} \) in Abb. 6.7 schaffen. Als Diagonalwert \( \delta_p \) ergibt sich nach Formel (6.2):

\[
\delta_p = m + (n + 11)
= n + m + 11
\]

Bleibt festzuhalten: Die Gleichheit der duplikatspezifischen Diagonalwerte \( \Delta \) und \( \delta \) sind notwendige Kriterien zur Redundanzerkennung; die alleinige Feststellung von (Sub-)Repeats bzw. (Sub-)Palindromen ist unzureichend, sie muss aber noch anschließend durchgeführt werden. Tabelle 6.2 gibt mit Vorgriff auf native_cluster_id - Attributwerte einen Auszug aus der DBT duplicates.repeats an. Die Akronyme / Kurzformen der Kopfzeilen bedeuten im

<table>
<thead>
<tr>
<th>CID</th>
<th>PID</th>
<th>DV</th>
<th>STARTPOS1</th>
<th>ENDPOS1</th>
<th>STARTPOS2</th>
<th>ENDPOS2</th>
<th>MIS</th>
<th>NC_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>92</td>
<td>2</td>
<td>929</td>
<td>19369472</td>
<td>19369756</td>
<td>19370401</td>
<td>19370685</td>
<td>-10</td>
<td>0</td>
</tr>
<tr>
<td>92</td>
<td>2</td>
<td>929</td>
<td>21309279</td>
<td>21309582</td>
<td>21310208</td>
<td>21310511</td>
<td>-8</td>
<td>0</td>
</tr>
<tr>
<td>92</td>
<td>2</td>
<td>929</td>
<td>22973104</td>
<td>22973420</td>
<td>22974033</td>
<td>22974349</td>
<td>-10</td>
<td>567</td>
</tr>
<tr>
<td>92</td>
<td>2</td>
<td>929</td>
<td>22973188</td>
<td>22973429</td>
<td>22974117</td>
<td>22974358</td>
<td>-7</td>
<td>567</td>
</tr>
<tr>
<td>92</td>
<td>2</td>
<td>929</td>
<td>23377453</td>
<td>23377756</td>
<td>23378382</td>
<td>23378685</td>
<td>-8</td>
<td>568</td>
</tr>
<tr>
<td>92</td>
<td>2</td>
<td>929</td>
<td>23377516</td>
<td>23377778</td>
<td>23378445</td>
<td>23378707</td>
<td>-10</td>
<td>568</td>
</tr>
<tr>
<td>92</td>
<td>2</td>
<td>929</td>
<td>23377582</td>
<td>23377877</td>
<td>23378511</td>
<td>23378716</td>
<td>-10</td>
<td>568</td>
</tr>
</tbody>
</table>

Tabelle 6.2: Auszug aus der DBT duplicates.repeats (maskiertes Chr. 22 von homo sapiens)
KAPITEL 6. DATENKONTROLLE & -AUFBEREITUNG

<table>
<thead>
<tr>
<th>Duplikattyp</th>
<th>real (s)</th>
<th>user (s)</th>
<th>sys (s)</th>
<th>%usr</th>
<th>%sys</th>
<th>%wio</th>
<th>%idle</th>
</tr>
</thead>
<tbody>
<tr>
<td>Palindrom</td>
<td>7519,40</td>
<td>0,04</td>
<td>0,09</td>
<td>1</td>
<td>1</td>
<td>42</td>
<td>56</td>
</tr>
<tr>
<td>Repeat</td>
<td>7614,84</td>
<td>0,05</td>
<td>0,10</td>
<td>1</td>
<td>1</td>
<td>42</td>
<td>56</td>
</tr>
</tbody>
</table>

Tabelle 6.3: Laufzeiten dia_value - Indexerstellung mit runstats

Einzelnen: carrier_id, param_id, dia_value, mismatches, native_cluster_id.
In den DBT duplicates.repeats und duplicates.palindromes geschieht die Redundanzerken-
nung über die Primärschlüssel repeat_id bzw. palindrome_id und das Attribut redundancy_id.
Gilt für die Attributwerte eines Tupels repeat_id = redundancy_id bzw. palindrome_id = re-
dundancy_id, ist das Tupel nicht - redundant. Im Fall der Ungleichheit gibt der Attributwert
von redundancy_id des redundanten Tupels den Primärschlüssel des überlappenden Dupli-
kats an.
Die Redundanzerkennung erfolgt ausschließlich über zwei SQL PL Prozeduren
duplicates.get_redundance_rep(<carrier_id>, <param_id>)
und
duplicates.get_redundance_pal(<carrier_id>, <param_id>)

unter Ausnutzung der spezifischen Diagonalwerte. Da diese später auch als 'Rausch-
filter' in der Webapplikation Verwendung finden, ist möglicherweise eine Indexerzeugung
vorteilhaft. db2advis kalkulierte hierbei jeweils eine knapp 17%-ige Verbesserung durch In-
dexerzeugung.
Das Anlegen der jeweils empfohlenen Indizes für Repeats und Palindrome dauerte inklusive
anschließendem runstats pro Duplikattyp gut zwei Stunden (s. Tabellen 6.3 und 6.4 auf).
Eine anschließende Neubewertung nach der Indexerstellung und runstats durch db2advis
ergab sogar noch deutlich bessere timerons - Werte.

| Duplikattyp | kalkulierte timerons ohne Index | kalkulierte timerons mit Index | Verbesserung | Anz. verwendeter Indexpages | Neube-
wertung timerons |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Palindrom</td>
<td>215.798</td>
<td>179.288</td>
<td>16.92%</td>
<td>≈ 673.000</td>
<td>19.218</td>
</tr>
<tr>
<td>Repeat</td>
<td>217.034</td>
<td>180.243</td>
<td>16.95%</td>
<td>≈ 670.000</td>
<td>19.318</td>
</tr>
</tbody>
</table>

Tabelle 6.4: timerons - Ergebnisse und Speicherplatzbelegung für beide Indizes

Trotz der Indexerzeugung vollzog sich die Änderung der Attributwerte von redundancy_id
auffällig langsam. Ein eigens auf Dateibasis funktionierendes C - Programm, redundancy_el-
minator, das redundante und nichtredundante Duplikate über eine verkettete Liste sowie
rekursivem mergesort (s. [S92]) trennt, und mittels

xld -O5 -q64 -o redundancy_eliminator redundancy_eliminator.c

#Version 6.0
4 auf 32bittigem AIX sollte man anschließend unbedingt noch /usr/ccs/bin/ldedit zur Stackvergrößerung
ausführen oder besser gleich mergesort iterativ implementieren
KAPITEL 6. DATENKONTROLLE & -AUFBEREITUNG

<table>
<thead>
<tr>
<th></th>
<th>real (s)</th>
<th>user (s)</th>
<th>sys (s)</th>
<th>%usr</th>
<th>%sys</th>
<th>%wio</th>
<th>%idle</th>
</tr>
</thead>
<tbody>
<tr>
<td>hs_masked_rep</td>
<td>204,69</td>
<td>0,32</td>
<td>1,18</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>77</td>
</tr>
<tr>
<td>hs_masked_pal</td>
<td>190,97</td>
<td>0,24</td>
<td>1,15</td>
<td>2</td>
<td>1</td>
<td>20</td>
<td>77</td>
</tr>
<tr>
<td>chimp_unmasked_rep</td>
<td>12,048</td>
<td>0,34</td>
<td>1,23</td>
<td>2</td>
<td>2</td>
<td>34</td>
<td>62</td>
</tr>
<tr>
<td>chimp_unmasked_pal</td>
<td>6,174</td>
<td>0,44</td>
<td>1,36</td>
<td>2</td>
<td>2</td>
<td>35</td>
<td>61</td>
</tr>
<tr>
<td>hs_unmasked_rep</td>
<td>27,624</td>
<td>0,49</td>
<td>1,18</td>
<td>1</td>
<td>1</td>
<td>26</td>
<td>71</td>
</tr>
<tr>
<td>hs_unmasked_pal</td>
<td>32,371</td>
<td>0,49</td>
<td>1,36</td>
<td>1</td>
<td>1</td>
<td>26</td>
<td>71</td>
</tr>
</tbody>
</table>

Tabelle 6.5: Laufzeiten der Redundanzerkennung durch SQLPL-Prozedur

kompiert wurde, ist mindestens um Faktor 10 (Basis: real-Werte von den Tabellen 6.5 und 6.6) schneller. Aussagekräftiger sind aber noch die %usr-Werte: Bei acht voll ausgelasteten Prozessoren (%usr + %sys = 100%) der Produktionsumgebung läuft ein Prozessor bei 100%: 8 = 12,5% unter voller Auslastung. Während die SQL PL Prozedur nur eine relative schlechte Auslastung (max. 4%) erreicht bei erheblichen I/O-Latenzzeiten (ca. 20%-30%), wird beim C-Programm eine wesentlich bessere Prozessorauslastung ([12,5%]) und erheblich geringere I/O-Latenzeit (max. 2%) erreicht.

Es ist also empfehlenswert, die Redundanzbereinigung vor einem Duplikat-LOAD über ein C-Programm oder sogar als VMATCH -selection function-Bibliothek vorzunehmen: Einerseits aus Laufzeitwirkungen und andererseits spart man das Attribut redundancy_id ein, welches hier überhaupt nur aus didaktischen Überlegungen eingeführt wurde.

Zwar sind nach den Tabellen C.8 auf Seite 135 bis C.13 auf Seite 140 i. d. R. jeweils 'nur' 1 oder 7 Prozent der Repeats oder Palindrome redundant, absolut gesehen entspricht dies knapp 9,35 Mio. zu entfernenden Datensätzen. Mit der Entfernung der redundanten Datensätze ging keine Freigabe von Tabellenbereichs- (Index und Daten) einher, offensichtlich verbleiben bei allen Seiten noch nicht - redundante Datensätze.

<table>
<thead>
<tr>
<th></th>
<th>real (s)</th>
<th>user (s)</th>
<th>sys (s)</th>
<th>%usr</th>
<th>%sys</th>
<th>%wio</th>
<th>%idle</th>
</tr>
</thead>
<tbody>
<tr>
<td>hs_masked_rep</td>
<td>7,30</td>
<td>5,67</td>
<td>0,49</td>
<td>10</td>
<td>1</td>
<td>2</td>
<td>87</td>
</tr>
<tr>
<td>hs_masked_pal</td>
<td>7,60</td>
<td>5,53</td>
<td>0,60</td>
<td>9</td>
<td>1</td>
<td>2</td>
<td>87</td>
</tr>
<tr>
<td>chimp_unmasked_rep</td>
<td>598,50</td>
<td>572,89</td>
<td>22,11</td>
<td>12</td>
<td>1</td>
<td>0</td>
<td>87</td>
</tr>
<tr>
<td>chimp_unmasked_pal</td>
<td>579,46</td>
<td>554,60</td>
<td>21,24</td>
<td>12</td>
<td>1</td>
<td>0</td>
<td>87</td>
</tr>
<tr>
<td>hs_unmasked_rep</td>
<td>1,961,41</td>
<td>1,886,31</td>
<td>67,19</td>
<td>12</td>
<td>1</td>
<td>0</td>
<td>87</td>
</tr>
<tr>
<td>hs_unmasked_pal</td>
<td>1,862,06</td>
<td>1,798,75</td>
<td>56,75</td>
<td>12</td>
<td>1</td>
<td>0</td>
<td>87</td>
</tr>
</tbody>
</table>

Tabelle 6.6: Laufzeiten der Redundanzerkennung durch C-Programm

6.2.2.2 Zusammenfassen sich teilweiser überlappende Repeats bzw. Palindrome durch das native clustering

In Abb. 6.8 auf Seite 72 sind zwei sich teilweise überlappende Repeats E mit $E_1[m \ldots m+6] \in B^7$, $E_2[n \ldots n+6] \in B^7$ und F mit $F_1[m+3 \ldots m+10] \in B^8$, $F_2[n+3 \ldots n+10] \in B^8$ dargestellt, die jeweils genau zwei mit Kleinbuchstaben symbolisierte Mismatches enthalten. Lassen sich E und F nicht zu genau einem Repeat $G_1 \in R$ mit $G_1[m \ldots m+10] \in B^{11}$, $G_2[n \ldots n+10] \in B^{11}$ und drei Mismatches zusammenfassen? Voraussetzung ist wiederum die Gleichheit der Diagonalwerte $\Delta_E$ und $\Delta_F$ von E und F:

$$\Delta_E = n - m$$
\[ \Delta F = (n + 3) - (m + 3) \]
\[ = n - m \]
\[ = \Delta E \]

Da neben der Erfüllung der notwendigen Voraussetzung auch noch \( m \leq m + 3 \leq m + 6 \leq m + 10 \) (sich überlappende Koordinaten von \( E_1 \) und \( F_1 \)) gilt, können \( E \) und \( F \) zu \( G \) verschmolzen werden.

Das Verschmelzen von Repeats bzw. Palindromen lässt sich wie folgt formalisieren:

**Definition 6.2 (Präfix, Suffix)** Sei \( D[g \ldots j] \in \mathcal{B}^{j-g+1} \) mit \( 1 \leq g \leq h < i \leq j \). Dann ist \( D[g \ldots h] \) ein Präfix von \( D \) und \( D[i \ldots j] \) ein Suffix von \( D \).

**Definition 6.3 (native Repeat, \( \mathcal{R}_N \))** Seien \( R_1 \in \mathcal{R} \) mit den Repeatarmen \( R'_1[m_1 \ldots m_1 + l_1] \in \mathcal{B}^{l_1+1} \) und \( R'_1[n_1 \ldots n_1 + l_1] \in \mathcal{B}^{l_1+1} \), \ldots, \( R_g \in \mathcal{R} \) mit den Repeatarmen \( R'_g[m_g \ldots m_g + l_g] \in \mathcal{B}^{l_g+1} \) und \( R'_g[n_g \ldots n_g + l_g] \in \mathcal{B}^{l_g+1} \).重复s mit \( m \in N^+, n \in N^+, 1 \leq k \leq g \) und \( \Delta R_k = \ldots = \Delta R_g \) gegeben. O. B. d. A. sei \( m_1 < \ldots < m_g \) und nach Definition 6.1 auf Seite 68 folglich \( n_1 < \ldots < n_g \).

\( \forall k \in N^+, 1 \leq k < g : m_k < m_{k+1} \leq m_k + l_k < m_{k+1} + l_{k+1} \Rightarrow R'_k[m_k \ldots m_k + l_k] = R'_{k+1}[m_{k+1} \ldots m_k + l_k] \), d. h., ein Suffix von \( R'_k(R_k) \) ist zugleich Präfix von \( R'_{k+1}(R'_{k+1}) \).

Dann ist \( R \in \mathcal{R} \) mit den Repeatarmen \( R'[m_1 \ldots m_g + l_g] \in \mathcal{B}^+ \) und \( R'[n_1 \ldots n_g + l_g] \in \mathcal{B}^+ \) ein native Repeat (von \( R_1, \ldots, R_g \)).

Die Menge aller native Repeats wird mit \( \mathcal{R}_N \) bezeichnet.

**Definition 6.4 (native Palindrom, \( \mathcal{P}_N \))** Seien \( P_1 \in \mathcal{P} \) mit den Palindromarmen \( P'_1[i_1 \ldots i_1 + l_1] \in \mathcal{B}^{l_1+1} \) und \( P'_1[j_1 \ldots j_1 + l_1] \in \mathcal{B}^{l_1+1} \), \ldots, \( P_h \in \mathcal{P} \) mit den Palindromarmen \( P'_{h}[i_h \ldots i_h + l_h] \in \mathcal{B}^{l_h+1} \) und \( P'_{h}[j_h \ldots j_h + l_h] \in \mathcal{B}^{l_h+1} \) Palindrome mit \( m \in N^+, n \in N^+, 1 \leq k \leq g \) und \( \delta P_1 = \ldots = \delta P_h \) gegeben. O. B. d. A. sei \( i_1 < \ldots < i_h \) und nach Definition 6.1 auf Seite 68 folglich \( j_1 < \ldots < j_h \).

\( \forall k \in N^+, 1 \leq k < h : i_k < i_{k+1} \leq i_k + l_k < i_{k+1} + l_{k+1} \Rightarrow P'_k[i_k \ldots i_k + l_k] = P'_{k+1}[i_{k+1} \ldots i_k + l_k] \), d. h., ein Suffix von \( P'_k(P_k) \) ist zugleich Präfix von \( P'_{k+1}(P_{k+1}) \).
Dann ist \( P \in \mathcal{P} \) mit den Palindromarmen \( P'[\mathcal{m}_1 \ldots \mathcal{i}_h + l_h] \in \mathcal{B}^+ \) und \( P''[\mathcal{n}_1 \ldots \mathcal{j}_h + l_h] \in \mathcal{B}^+ \) ein native Palindrome (von \( P_1, \ldots, P_h \)).

Die Menge aller native Palindrome wird mit \( \mathcal{P}_N \) bezeichnet.

Aus Datenbanksicht sind zunächst einmal zusammengehörende, d. h. sich teilweise überlappende und mit identischem Diagonalwert ausgestattete Repeats bzw. Palindrome zu ermitteln und zu vermerken.

Trotz dem von \textit{db2advis} empfohlenen und angelegten Index für den \textit{dia_value} bei der Redundanzerkennung, fand eine erneute Indexüberprüfung durch \textit{db2advis} statt. Erstaunlicherweise schlug \textit{db2advis} einen im Vergleich zum bereits verfügbaren Index der Redundanzerkennung (fast) identischen Index vor. Erst nach einem modifizierten und manuell ausgeführten \textit{runstats} - Aufruf, stellte \textit{db2advis} fest, dass die bereits angelegten Indizes für Repeats und Palindrome optimal sind. Dies legt den Schluss nahe, dass durch den vorherigen \textit{runstats} - Aufruf die Indexaktualisierung nicht (in vollem Umfang) durchgeführt wurde. Die letztendlich ermittelten timerons - Werte (Repeats: 19.260,5353; Palindrome: 19.161,1426) entsprechen denen der Neubewertung der in Tabelle 6.4 auf Seite 70 angegebenen timerons - Werte.

Beim \textit{LOAD} wird standardmäßig für die Repeat- und Palindromdaten ein Nullwert geladen. Duplikate, die zu ein- und demselben Cluster gehören, lassen sich über den gemeinsamen Attributwert von \textit{native_cluster_id} identifizieren. Duplikate, die sich nicht mit anderen Duplikaten clustern lassen, erhalten als Kennzeichnung den festen Attributwert 0.

Für die Umsetzung des Markierens und die davon partiell abhängige Interpretation der Attributwerte standen mehrere Alternativen zur Auswahl. Für die Zuweisung eines für jedes native cluster duplikatweit eindeutigen Attributwertes kommt eine

1. \texttt{CREATE SEQUENCE} - Konstruktion
2. separate Cluster_ID - Verwaltung
3. Abfrage nach der max. Cluster_ID

in Betracht. Zu (1) siehe u. a. \cite{BW03}. (3) wird wegen der langen Laufzeit über alle Duplikatdatensätze zur Bestimmung des momentan maximalen \textit{native_cluster_id} - Attributwertes verworfen. Realisiert wurde (2), indem der momentan maximale Attributwert in einer separaten DBT verwaltet wird. Gegenüber (1) bietet (2) keine relevanten Vorteile, sodass beide Implementierungen gleichberechtigt sind. Dieser Zusatzaufwand ist generell verzichtbar, wenn man auf die duplikatweit eindeutigen Attributwerte verzichtet und die Eindeutigkeit auf Duplikate mit gleicher \textit{carrier_id} und \textit{param_id} reduziert. Dem Vorteil der etwas einfacheren Realisierung stehen aber später u. U. komplexere Abfragen gegenüber, auch wenn diese Verarbeitungsschritte singulär sind.

Zum Setzen der \textit{native_cluster_id} - Attributwerte wurden zwei SQL PL Prozeduren

\[
duplicates.set_rep\_nat\_cl\_id(<\textit{carrier_id}>, <\textit{param_id}>)
\]

bzw.

\[
duplicates.set_pal\_nat\_cl\_id(<\textit{carrier_id}>, <\textit{param_id}>)
\]

geschrieben. Die obligatorischen Parameterangaben \textit{carrier_id} und \textit{param_id} sollen eine schrittweise Abarbeitung erzwingen.

Trotz optimalem Index und Indexaktualisierung liegen die in Tabelle 6.7 auf Seite 74 festgehaltenen Laufzeiten im Bereich der Laufzeiten der Redundanzerkennung (und sind ebenfalls
KAPITEL 6. DATENKONTROLLE & -AUFBEREITUNG

<table>
<thead>
<tr>
<th></th>
<th>real (s)</th>
<th>user (s)</th>
<th>sys (s)</th>
<th>%usr</th>
<th>%sys</th>
<th>%wio</th>
<th>%idle</th>
</tr>
</thead>
<tbody>
<tr>
<td>hs_masked_rep</td>
<td>164,58</td>
<td>0.39</td>
<td>1,05</td>
<td>1</td>
<td>2</td>
<td>21</td>
<td>77</td>
</tr>
<tr>
<td>hs_masked_pal</td>
<td>151,62</td>
<td>0.43</td>
<td>1,11</td>
<td>1</td>
<td>2</td>
<td>20</td>
<td>77</td>
</tr>
<tr>
<td>chimp_unmasked_rep</td>
<td>11.659,10</td>
<td>0.34</td>
<td>1,29</td>
<td>1</td>
<td>0</td>
<td>23</td>
<td>76</td>
</tr>
<tr>
<td>chimp_unmasked_pal</td>
<td>10.663,20</td>
<td>0.35</td>
<td>1,28</td>
<td>2</td>
<td>0</td>
<td>23</td>
<td>75</td>
</tr>
<tr>
<td>hs_unmasked_rep</td>
<td>37.321,83</td>
<td>0.50</td>
<td>1,23</td>
<td>13</td>
<td>1</td>
<td>28</td>
<td>57</td>
</tr>
<tr>
<td>hs_unmasked_pal</td>
<td>15.799,98</td>
<td>0.34</td>
<td>1,35</td>
<td>14</td>
<td>2</td>
<td>38</td>
<td>46</td>
</tr>
</tbody>
</table>

Tabelle 6.7: Laufzeiten der Redundanzerkennung durch C - Programm

Die beiden zweistelligen %usr-Werte sind durch einen anderen, voll ausgelasteten Prozessor, verursacht. Bei diesem Datenumfang ist analog zur Redundanzerkennung und -beseitigung die Durchführung des native clustering vor jeglichem LOAD durch eine effiziente Applikation empfehlenswert: Zum einen verbessern sich aufgrund der beträchtlichen Tupelreduzierung die LOAD-Laufzeiten und zum anderen kann dann auf das native_cluster_id-Attribut verzichtet werden. Die Zeit zur Entwicklung dieser Applikation wird sukzessive über die Zeitersersparnis der Applikationsausführung überversponiert.

Mit Hilfe des native clustering lassen sich, wie man den Abb. 6.11 und 6.12 auf Seite 77 entnehmen kann, zum einen viele Millionen Duplikatdatensätze einsparen und zum anderen i. a. wesentlich signifikantere Duplikate finden: Rund 40,5 Millionen Repeats (39,90 Millionen Palindrome) lassen sich zu rund 12,84 Millionen native Repeats (12,80 Millionen native Palindrome) zusammenfassen. Am konkreten Beispiel bedeutet dies, dass die jeweils rund 1,6 Millionen Duplikate von Chromosom 8 in den Abbildungen 6.11 und 6.12 auf je 0,5 Millionen native Duplikatcluster reduziert werden können. Es ist frappierend, festzustellen, wie sehr sich beide Abb. ähneln, auch wenn die zugrunde liegenden Tabellen C.14 auf Seite 141 bis C.19 auf Seite 146 Unterschiede aufweisen.

Bei den Repeats besteht u. a. ein native Cluster aus 171 (Palindrome: 62) Einzelduplikaten (s. Tabelle C.14 auf Seite 141 und C.17 auf Seite 144). Das längste native Repeatcluster (Palindromecluster) setzt sich aus 60.647 (100.520) Bp zusammen (s. Tabelle C.14 und C.17).

Die durchschnittlichen Duplikatarmlängen aller ursprünglichen Duplikate, die sich zu einem native cluster (native_cluster_id > 0) verschmelzen lassen, sind mit ‘+’ bzw. ‘*’ markiert. Die durchschnittlichen Duplikatarmängen der daraus zusammengesetzten native Duplikatcluster sind mittels ‘x’ und ‘□’ abgebildet. Die homogenen Ergebnisse sind mit der sensitiven VMATCH - Parameterwahl erkliarbar. Ferner fällt auf, dass neben den Chromosomen 16, 17 und 22 insbesondere Y eine Ausnahme darstellt, vor allem mit Bezug auf Abb. 6.10. Davon unabhängig ist die Steigerung der durchschnittlichen Duplikatarmängen um jeweils ca. 50%. Erfreulich ist zudem die Steigerung der maximalen Duplikatarmängen um durchschnittlich 20% - 40% und in Ausnahmefällen sogar bis über 250%.

Im Sinne eines vereinfachten Datenzugriffs für die nachfolgenden Operationen werden die nativen Duplikatcluster in eine separate DBT - jeweils für Repeats und Palindrome - versetzt und die zugrunde liegenden (Original-)Duplikate gelöscht.

6 von den beiden unmaskierten Genomen und dem maskierten Genom von homo sapiens
7 s. Tabellen C.14 und C.17 auf Seite 144
Diese beiden DBT, `results.repeats_native` und `results.palindromes_native`, entsprechen dem Schema ihrer korrespondierenden DBT (s. Abschnitt 4.2.3.1 auf Seite 43) mit folgenden Abweichungen:

- kein `evalue`-Attribut: Es steht kein Programm zur Berechnung des neuen Evalues zur Verfügung.
- kein `redundance_id`-Attribut: Die Redundanzbereinigung erfolgte bereits und kann darüber hinaus nicht mehr vorkommen.
- kein `native_cluster_id`-Attribut: Dieses Attribut ist nicht (mehr) notwendig.
- zusätzliches `num_parts`-Attribut: Gibt an, aus wie vielen ursprünglichen Duplikaten sich dieses native Duplikatchuster zusammensetzt.


### 6.2.2.3 Neubestimmung der Anzahl Mismatches

Für die letzten fünf der in Tabelle 6.2 auf Seite 69 angegebenen Repeats ergeben sich die in Tabelle 6.8 angegebenen native Repeatcluster (NP bedeutet `num_parts`).

Die Bestimmung der Anzahl von Mismatches ist i. A.8 nicht ableitbar, sodass die Anzahl von Mismatches eines geclusterten Repeats nachträglich über Sequenzvergleiche ermittelt werden muss. In den Abb. 6.13 auf Seite 78 und 6.14 auf Seite 79 sind jeweils zwei sich überlappende Repeatarme $E'_1, F'_1, E''_1, F''_1 \in B^+$ angegeben, die sich zu $G'_1 \in B^+$ bzw. $G''_1 \in B^+$ vereinen lassen. Obwohl in beiden Abb. jeder Repeatarm zehn Mismatches (‘X’) aufweist, enthält $G'_1$ zwanzig Mismatches und $G''_1$ nur elf Mismatches. Da wie gesehen die Verteilung der Mismatches im Prinzip durch die Repeatdaten allein nicht bestimbar ist, bedarf es der zwingenden Heranziehung der DNA-Sequenzen. Der geschilderte Sachverhalt gilt analog auch für die Palindrome.

Das Setzen der richtigen Mismatchanzahl übernehmen zwei SQL-PL-Prozeduren

---

<table>
<thead>
<tr>
<th>CID</th>
<th>PID</th>
<th>DV</th>
<th>STARTPOS1</th>
<th>ENDPOS1</th>
<th>STARTPOS2</th>
<th>ENDPOS2</th>
<th>MIS</th>
<th>NP</th>
</tr>
</thead>
<tbody>
<tr>
<td>92</td>
<td>2</td>
<td>929</td>
<td>22973104</td>
<td>22973429</td>
<td>22974033</td>
<td>22974358</td>
<td>567</td>
<td>2</td>
</tr>
<tr>
<td>92</td>
<td>2</td>
<td>929</td>
<td>23377453</td>
<td>23377787</td>
<td>23378382</td>
<td>23378716</td>
<td>568</td>
<td>3</td>
</tr>
</tbody>
</table>

Tabelle 6.8: Auszug aus der DBT `duplicates.repeats_native` (maskiertes Chr. 22 von *homo sapiens*)
KAPITEL 6. DATENKONTROLLE & -AUFBEREITUNG

Abbildung 6.9: Vergleich der ∅ Repeatarmlängen mit und ohne native Clustering

Abbildung 6.10: Vergleich der ∅ Palindromarmlängen mit und ohne native Clustering
KAPITEL 6. DATENKONTROLLE & -AUFBEREITUNG

Abbildung 6.11: Native Repeatclusterstatistik

Abbildung 6.12: Native Palindromclusterstatistik
KAPITEL 6. DATENKONTROLLE & -AUFBEREITUNG

Abbildung 6.13: Mögliche Mismatchverteilung (’x’) bei überlappenden Repeats

\[
\begin{align*}
\text{results.set_rep_nat_mis}(\text{<carrier_id>, <param_id>}) \\
\text{und} \\
\text{results.set_pal_nat_mis}(\text{<carrier_id>, <param_id>})
\end{align*}
\]

Durch die Wiederverwendung von in C implementierter Sequenzfunktionen (s. Abschnitt 6.2.1 auf Seite 65) erfolgt das Eintragen bei guter Prozessorauslastung relativ schnell (s. Tabelle 6.9). Für die höhere Prozessorauslastung bei den unmaskierten Sequenzen ist wohl die reverse complement Bestimmung verantwortlich.

Im Zuge der Mismatcheintragung erfolgt die automatische Mitberechnung der prozentualen Identität (\textit{prcnt_ident}), die durchaus auch geringfügig unter das Ausgangsniveau (hier 99% und 95%) sinken kann. Ein Löschen solcher Datensätze bietet sich nicht an, da der Informationsgewinn gegenüber der strikten Einhaltung von Kriterien i. A. überwiegt.

6.2.2.4 Erweitertes Zusammenfassen von Repeats bzw. Palindromen durch das extended native clustering

Die Voraussetzung, dass sich Duplikate für das native clustering (teilweise) überlappen müssen, ist, wie ausführlich geschildert, ein außerordentlich pragmatisches und rigoroses Entscheidungskriterium, aber auf Duplikate mit geringem Duplikat(arm)abstand nicht anwendbar. Es ist wünschenswert und sinnvoll, darüber hinaus Duplikate mit gleichem Diagonalwert

\[
\begin{array}{|c|c|c|c|c|c|c|}
\hline
\text{real (s)} & \text{user (s)} & \text{sys (s)} & \%\text{us} & \%\text{sys} & \%\text{wio} & \%\text{idle} \\
\hline
\text{hs\_masked\_rep} & 39,15 & 0,36 & 1,07 & 2 & 1 & 10 & 87 \\
\text{hs\_masked\_pal} & 87,00 & 0,35 & 1,12 & 1 & 1 & 18 & 81 \\
\text{chimp\_unmasked\_rep} & 784,38 & 0,34 & 1,10 & 5 & 1 & 7 & 87 \\
\text{chimp\_unmasked\_pal} & 526,80 & 0,29 & 1,29 & 9 & 1 & 16 & 74 \\
\text{hs\_unmasked\_rep} & 1979,86 & 0,34 & 1,04 & 7 & 1 & 12 & 80 \\
\text{hs\_unmasked\_pal} & 1769,60 & 0,31 & 1,27 & 12 & 3 & 40 & 45 \\
\hline
\end{array}
\]

Tabelle 6.9: Laufzeiten zum Setzen der neuen Mismatchanzahl

78
KAPITEL 6. DATENKONTROLLE & -AUFBEREITUNG

Abbildung 6.14: Mögliche Mismatchverteilung ('x') bei überlappenden Repeats

und geringem Duplikatabstand gleichfalls zu clustern. Da neben dem Duplikatabstand auch die Duplikat(arm)länge von entscheidender Bedeutung ist, kristallisiert sich die prozentuale Identität \( \text{prcnt\_ident} \) als tragfähiges Entscheidungskriterium heraus: Wenn die prozentuale Identität des verschmolzenen Duplikats mindestens so hoch ist wie die prozentuale Identität eines der / beider zugrunde liegenden Duplikate, wird es gleichfalls geclustert.

Datengrundlage sind sowohl die Repeats und native Repeats als auch die Palindrome und native Palindrome, deren Ergebnisse jeweils in einer separaten DBT abgespeichert werden. Das dafür überhaupt Anwendungsfälle existieren belegt Abb. 6.15. Dort sind die beiden Repeatarme \( I_1 \in B_{200} \) und \( J_1 \in B_{200} \) der Repeats \( I, J \in R \) mit jeweils 10 Mismatches (\( \equiv 95\% \) Identität) abgebildet. Zwischen \( I_1 \) und \( J_1 \) liegen 40 Bp (Repeat(arm)abstand), darunter aber nur zwei Mismatches ('X'). Fasst man \( I_1 \) und \( J_1 \) inklusive Repeatarmabstand zusammen, ergibt sich ein Repeatarm von 200 + 200 + 40 = 440 Bp, der 10 + 10 + 2 = 22 Mismatches (ergo \( \equiv 95\% \) Identität) aufweist und signifikanter ist als \( I \) bzw. \( J \). Nebenbei stellt sich wie beim vorangegangenen native clustering erneut eine Datenreduktion bei verbessertem Informationsgehalt ein. Läge aber zwischen \( I_1 \) und \( J_1 \) genau ein Mismatch, so würde das zusammengesetzte Repeat ((200 + 1 + 200 = 401 Bp) / (10 + 1 + 10 = 21 Mismatches) \( \approx 94,763\% \) Identität) das Kriterium knapp verfehlen. Diese 'Unzulänglichkeit' des extended native clusterings lässt sich aber grundsätzlich nicht beheben.

Abbildung 6.15: Anwendungseispiel zum extended native clustering

79

Beiden Ansätzen gemein ist die jeweilige synchrone Mismatchberechnung. Falls sich Repeats oder Palindrome clustern lassen, ist im Gegensatz zum native clustering keine a posteriori Mismatchbestimmung (s. 6.2.2.3 auf Seite 75) notwendig.

Aus Zeitnot und zu erwartender Laufzeitkomplexität wegen der immer noch hohen Duplikatanzahl, wird das extended native clustering im Rahmen dieser Diplomarbeit nicht realisiert.

6.2.2.5 Kennzeichnung nichtredundanter (Sub-)Repeats bzw. (Sub-)Palindrome

Wie eingangs von Abschnitt 6.2.2 auf Seite 66 angedeutet, dient das Setzen von overlap_id-Attributwerten vor allem zur Filterung der Duplikate bei grafischen Ausgaben. Sollen z.B. alle native Duplikatcluster vom unmaskierten X-Chromosom eingezeichnet werden, so sind dies nach den Abb. 6.11 auf Seite 77 und 6.12 auf Seite 77 jeweils rund 1,5 Mio. Datensätze. Da jeder Datensatz aus vier Koordinaten und i.d.R. 8-9 Ziffern enthält, setzt sich die Eingabedatei für die Bilderzeugung aus ca. 3.000.000•4•9 ≈ 100 MB zusammen.


Durch vier hier nicht näher angegebene SQL PL Prozeduren, jeweils für Repeats und Palindrome und (nicht)-clustering unterteilt fand das Setzen der overlap_id-Attributwerte statt, wobei vorher für die native Duplikatcluster noch optimale Indizes erzeugt wurden. Die semantische Codierung der Attributwerte gleicht jener zur Interpretation der redundance_id-Attributwerte: Gilt Attributwert des Primärschlüssels ≠ overlap_id-Attributwert, so enthält overlap_id den Attributwert des Primärschlüssels des vollständig überlappenden Duplikats. Hilfreich ist die Datenreduktion aufgrund des native clustering, da die Verarbeitung der Datensätze jetzt ausschließlich chromosomenweise (<carrier_id> (vorer <carrier_id>, <param_id>)) erfolgen muss.

Algorithmisch entspricht der Prozess der der Redundanzerkennung: Duplikate aufsteigend nach Diagonalwert und erster Startposition (startpos1) sortieren und dann anhand der Koordinaten prüfen, ob das aktuelle Duplikat in einem vorherigen liegt. Setze dann overlap_id-Attributwerte wie oben beschrieben.

Die Laufzeiten sind in der Tabelle 6.10 auf Seite 81 festgehalten.

Als Quintessenz ist anzumerken, dass sich Chromosom etwa 28% bis 42% der Datensätze bei einer grafischen Ausgabe einsparen lassen. Im Übrigen gibt es trotz unterschiedlicher VMATCH - Parameterwahl chromosomenweise einige vollkommen identische Datensätze. Es ist anzunehmen, dass diese Gleichheit durch ein nachfolgendes Gap verursacht wird, sodass eine Duplikatverlängerung im Rahmen der vorgegebenen VMATCH -
KAPITEL 6. DATENKONTROLLE & -AUFBEREITUNG

<table>
<thead>
<tr>
<th></th>
<th>real (s)</th>
<th>user (s)</th>
<th>sys (s)</th>
<th>%usr</th>
<th>%sys</th>
<th>%wio</th>
<th>%idle</th>
</tr>
</thead>
<tbody>
<tr>
<td>hs_masked_rep</td>
<td>86,38</td>
<td>0,08</td>
<td>0,69</td>
<td>1</td>
<td>4</td>
<td>65</td>
<td>30</td>
</tr>
<tr>
<td>hs_masked_pal</td>
<td>80,53</td>
<td>0,14</td>
<td>0,58</td>
<td>2</td>
<td>4</td>
<td>67</td>
<td>27</td>
</tr>
<tr>
<td>chimp_unmasked_rep</td>
<td>11.567,45</td>
<td>0,23</td>
<td>0,72</td>
<td>1</td>
<td>0</td>
<td>26</td>
<td>73</td>
</tr>
<tr>
<td>chimp_unmasked_pal</td>
<td>11.118,49</td>
<td>0,30</td>
<td>0,54</td>
<td>1</td>
<td>0</td>
<td>26</td>
<td>73</td>
</tr>
<tr>
<td>hs_unmasked_rep</td>
<td>32.547,55</td>
<td>0,26</td>
<td>0,75</td>
<td>1</td>
<td>2</td>
<td>32</td>
<td>65</td>
</tr>
<tr>
<td>hs_unmasked_pal</td>
<td>18.928,93</td>
<td>0,17</td>
<td>0,72</td>
<td>1</td>
<td>2</td>
<td>12</td>
<td>85</td>
</tr>
</tbody>
</table>

Tabelle 6.10: Einige Laufzeiten zum Setzen der overlap_id - Attributwerte

Parameter verhindert wird.

6.3 Kapitelzusammenfassung und Fazit

Zu Beginn der Diplomarbeit war diese (umfangreiche) Datenkontrolle und -aufbereitung bestenfalls ansatzweise eingeplant. Erst nach und nach zeichnete sich das ganze Ausmaß an diesem beträchtlichen - aber quasi unverzichtbaren und zuweilen auch pragmatischen - Mehraufwand ab. Durch den Erkenntnisgewinn der Abarbeitung eines Vorverarbeitungsschrittes lies sich dieser mit Modifikationen manchmal auch anderweitig wiederverwenden. In Anbetracht der dadurch verursachten Auflösungsscheinerungen der ursprünglichen Konzeption dieser Diplomarbeit lässt sich zur Datenkontrolle und -aufbereitung folgendes Fazit ziehen:

- inhaltlich
  - Die Bestätigung der Korrektheit aller Duplikate in Bezug auf die DNA-Sequenzen stellt die Untersuchungen auf ein stabiles Fundament.
  - Die Redundanzbereinigung behebt methodische Unzulänglichkeiten.
  - Das native clustering reduziert den Datenumfang in beachtlichem Maße und verbessert gleichzeitig den Informationsgehalt.

- methodisch
  - Das ‘Stolpern’ von einem Prozessschritt zum anderen zog meistens umfangreiche und zeitaufwendige Änderungen am Datenbankschema und den Daten nach sich. Einige dieser Prozessschritte sind entbehrlich, die Redundanzbereinigung ist prinziell unverzichtbar und das native clustering sehr hilfs- und aufschlußreich.
  - Die Redundanzbereinigung und ggf. das native clustering sollten bei diesen Datenumfang unbedingt vor der Datenbankeinstellung der Daten erledigt werden.

Ansatzpunkte zu Laufzeitverbesserungen der Datenbankmethoden sind vielleicht Exklusiververbindungen (*connect to synteny in exclusive mode*) oder das sperren von Datenbankobjekten (*lock table duplicates,<duplicatetype>*). Wegen der signifikanten Datenreduzierung
ist ggf. eine Konfigurationsüberprüfung mit *autoconfigure* empfehlenswert. Sobald sich der Datenbestand einem 'finalen' Zustand nähert, sind Laufzeitmessungen mit *db2batch* vermutlich deutlich informativer im Hinblick auf die Datenbankkonfiguration. Der Großteil der Datenaufbereitung ist damit erfolgreich abgeschlossen. Im Folgekapitel steht aber erst die semantische und statistische Auswertung der (verbliebenen) Duplikatergebnisse an.
Kapitel 7

Statistische & funktionelle Duplikatdatenauswertung


7.1 Die DB HS_CORE und FED


Für einen gemeinsamen Datenzugriff auf SQL-Ebene wurde die föderierte DB FED erzeugt. Da sie hauptsächlich den logischen Zugriff auf die DB von SYNTENY und HS_CORE herstellt und selbst nur geringe Datennengen beherbergt, sind die Standard-SMS-Tabellenbereichs für diese Aufgabe prädestiniert.

1 und im Folgenden als HS_CORE tituliert wird
KAPITEL 7. STATISTISCHE & FUNKTIONELLE DUPLIKATDATENAUSWERTUNG

Der Vorgang der Erstellung einer fäderierten DB ist z. B. in [IBM V] nachlesen. Anzumerken ist, dass bei der Wahl eines falschen Servertyps (bspw. db2/aix anstelle von db2/6000) beim logischen Zugriff von FED auf SYNTENY oder HS_CORE die Instanz beendet wird. Im Kontext dieser Diplomarbeit ist es ausreichend zu wissen, dass die (physischen) DB von ENSEMBL und SYNTENY über einen Aliasnamen (nickname) logisch zugreifbar gemacht werden. Je nach Herkunft wird ihnen in FED entweder der Schemaname ensembl oder synten-
ny vorangestellt. Für die anstehenden Untersuchungen neu anzulegende Datenbankobjekte (DBT, Views ...) werden (physisch) bei der DB FED hinterlegt, um eine Vermischung von 'Original' - und 'Hilfs' - Datenbankobjekten zu verhindern.

Der Mehraufwand durch die semantische Aufteilung und physische Trennung der DB (hier SYNTENY und ENSEMBL) sowie das logische Zusammenfassen in einer weiteren DB (FED) lohnt sich aber bei Erweiterungen: So ließe sich hier noch ohne Einbußen an Übersichtlichkeit die DB von pan troglodytes hinzufügen. Ausgangsbasis für die Untersuchungen sind neben den Duplikat - DBT von SYNTENY die folgenden DBT von ENSEMBL:

- (1) basic.seq_region
- (2) basic.karyotype
- (3) basic.gene
- (4) basic.gene_stable_id
- (5) basic.exon
- (6) basic.exon_stable_id

(1) speichert Informationen u. a. zu Chromosomen. Die DBT wird für die Zuordnung von zytogenetischen Banden, Genen und Exons auf Chromosomen benötigt. (2) beinhaltet Angaben zu verschiedenen einfärbbaren Abschnitten (zytogenetischen Banden) eines Chromosoms. Aus diesen Angaben lassen sich auch die Koordinaten der Chromosomenarme ermitteln. (3) enthält neben der Zuordnung zum Chromosom vor allem Genkoordinaten samt Stranglokalisation. (4) vermerkt einige Zusatzinformationen zu (3). (5) und (6) vermerken im Wesentlichen Informationen analog zu (3) und (4).

7.2 Vorbereitungen zu den Untersuchungen

Das Hauptanliegen der statistischen Untersuchungen ist die Auswertung der Duplikatdaten nach topologischen Gesichtspunkten. Die topologischen Ebenen sind in Abb. 7.1 auf Seite 85 nach zunehmender Feingranularität angeordnet.

Zu beachten ist, dass (B), (C) und (D) eine Untergliederung von (A) nach diversen Kri-
terien sind, d. h., i. d. R. ist (C) ((D)) auch eine Untergliederung von (B) ((C)), aber es gibt Ausnahmen: Z. B. existieren Gene, die je zwei benachbarte zytogenetische Banden teilweise überlappen. Lediglich (E) ist eine strikte Untergliederung von (D) ohne Ausnahmefälle.

7.2.1 Bestimmung intragenischer Abschnitte

Die Koordinaten der Exons zur Bestimmung der intragenetischen Abschnitte von (E) stehen in der DBT basic.exon. Über eine geeignete Datentransformation per SQL unter Einbezie-
hung der DBT basic.gene sollte eine Aufbereitung und Extrahierung der Exondaten erfolgen.

3ENSEMBL klassifiziert die DBT in fünf Kategorien fundamental, feature and analysis, ID mapping, external references, miscellaneous, sodass hier für erstgenannter der Schemaname basic vorangestellt ist.
Abbildung 7.1: Schema der topologischen Ebenen


7.2.2 Definition und Bestimmung (inter-)genischer Abschnitte

Das in der DBT basic.gene Genkoordinaten enthalten sind, bedeutet noch nicht, dass man damit auch die (inter-)genischen Abschnitte kennt, da sich manche Gene ganz oder teilweise überlappen. Ein genischer Abschnitt - hier als (gene stretch) bezeichnet - ist hier entweder ein einzelnes nicht - überlappendes Gen, der maximale Ausdehnungsbereich von sich jeweils überlappenden Genen oder der zusammengefasste Bereich von zwei unmittelbar angrenzenden Genen. Die verbliebenen, genfreien Abschnitte werden hier als intergenische Abschnitte definiert. Eine SQL PL Prozedur namens get.gene_stretches(<sequence_id>) bestimmt auf Grundlage von basic.gene die genischen Abschnitte inklusive Koordinaten und fügt diese in der DBT ensembl.gene_stretches ein. Beim Einfügen wird ein positiver Attributwert für den Primärschlüssel gene_stretch_id vergeben. Ein Trigger stellt erstens die Lückenlosigkeit dieser Attributwerte und zweitens die Monotonie sicher. D. h., wenn ((<s_i>, <e_i>) die Koordinaten des genischen Abschnitts A_i sind und ((<s_{i-1}>, <e_{i-1}>) ((<s_{i+1}>, <e_{i+1}>)) die des unmittelbar vorherigen (folgenden) Abschnitts, so gilt A_{i-1} = A_{i-1} (A_i + 1 = A_{i+1}) und s_{i-1} ≤ e_{i-1} < s_i ≤ e_i < s_{i+1} ≤ e_{i+1}.

Um mit Hilfe der SQL PL Prozedur ensembl.get.intergene_stretches und enssembl.gene_stretches die DBT ensembl.intergene_stretches zu füllen, ist noch die DBT basic.seq_region hinzuziehen, die die Chromosomenlänge bereitstellt. Auch hier kontrolliert ein Trigger die Lückenlosigkeit und Monotonie der Attributwerte des Primärschlüssels intergene_stretch_id, mit dem Unterschied, dass diese alle negativ sind.

Alle Datensätze von enssembl.gene_stretches und ensembl.intergene_stretches werden in der DBT ensembl.gene_intergene_stretches vereint, die aus den Attributen

- stretch_id (PK)
- carrier_id
- stretch_start

4D. h. zwischen beiden Genen liegt kein intergenischer Abschnitt.
5Auch wenn der erste (letzte) genetische Abschnitt eines Chromosoms keinen Vorgänger (Nachfolger) hat.
6Die Beweggründe werden weiter unten erläutert.

85
• stretch_end
• stretch_length
• num_genes

Über die DB FED werden die entsprechenden stretch_id - Attributwerte in startpos1_func_id, endpos1_func_id, startpos2_func_id, endpos2_func_id eingetragen (s. Abschnitt 4.2.3.1 auf Seite 43) und damit auch num_func. Nach diesem Vermerk in den Duplikatdatensätzen ver- einfacht sich die Auswertung methodisch.

7.3 Qualitative Einschätzung der Duplikatdaten

Als übliche Indikatoren für die Güte eines Duplikats verwendet man die prozentuale Übereinstimmung der Duplikatarmsequenzen und / oder den Erwartungswert (evalue). Beiden Kennzahlen liegt jedoch eine Berechnung auf Basis einer konkreten DNA- Sequenz zugrunde. Die hier vorzunehmende qualitative Einschätzung soll sich auf quantitative und / oder statistische Kenngrößen beschränken.
Der mathematische Ansatz in Abschnitt 3.3 auf Seite 27 hilft nicht weiter. Die daraus zu ziehende Erkenntnis, dass ein Duplikat i. A. mit zunehmender Duplikatarmlänge signifikanter wird, lässt sich in einem anderen Ansatz ausnutzen, indem man die Duplikate in Abhängigkeit ihrer Duplikatarmlänge gruppiert und durchzählt. Mit zunehmender Duplikatarmlänge sinkt die Anzahl der Duplikate tendenziell, aber diese sind prinzipiell signifikanter.
Die dadurch bedingte sukzessive Separation von nicht bzw. weniger signifikannten (‘verrauschten’) Duplikaten zu den besonders signifikannten ist mathematisch wie folgt beschreibbar:
Seien $l_1, \ldots, l_m$ die (eindeutigen) Duplikat(arm)längen der nach carrier_id und param_id aufgeteilten Duplikate. Sei $n_k$ ($1 ≤ k ≤ m$) die Anzahl der Attribute mit Länge $l_k$. Dann berechnet sich die Anzahl $s_k$ ($1 ≤ k ≤ m$), aller Duplikate mit maximaler Duplikatarmlänge nach:

$$s_k := \sum_{i=1}^{k} n_k$$

Die Tabelle 7.1 auf Seite 87 enthält einige Palindrombeispieldaten von Chromosom 1 (carrier_id = 1 und param_id = 3).
Wie in Abb. 7.2 auf Seite 87 dargestellt, berechnet sich der Anstieg $\alpha_k$ (in Grad) der Strecke A nach der Formel

$$\alpha_k = \tan^{-1} \left( \frac{s_{k+1}-s_k}{l_{k+1}-l_k} \right)$$

7 gilt nur falls entweder endpos1_func_id < 0 und startpos2_func_id < 0 oder endpos1_func_id > 0 und startpos2_func_id > 0.
KAPITEL 7. STATISTISCHE & FUNKTIONELLE DUPLIKATDATENAUSWERTUNG

Tabelle 7.1: Beispieldaten vom unmaskierten humanen Chromosom 1

<table>
<thead>
<tr>
<th>CID</th>
<th>PID</th>
<th>$k$</th>
<th>$l_k$</th>
<th>$n_k$</th>
<th>$s_k$</th>
<th>$l_{k+1} - l_k$</th>
<th>$s_{k+1} - s_k$</th>
<th>$\alpha_k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>1</td>
<td>100</td>
<td>47518</td>
<td>47518</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>2</td>
<td>101</td>
<td>44500</td>
<td>92018</td>
<td>1</td>
<td>44500</td>
<td>89.99</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>3</td>
<td>102</td>
<td>39610</td>
<td>131628</td>
<td>1</td>
<td>39610</td>
<td>89.99</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>4</td>
<td>103</td>
<td>45680</td>
<td>177308</td>
<td>1</td>
<td>45680</td>
<td>89.99</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>5</td>
<td>104</td>
<td>36444</td>
<td>213752</td>
<td>1</td>
<td>36444</td>
<td>89.99</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>530</td>
<td>5806</td>
<td>1129513</td>
<td>608</td>
<td>1</td>
<td>0.09</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>531</td>
<td>6031</td>
<td>1129514</td>
<td>225</td>
<td>1</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>532</td>
<td>6528</td>
<td>1129515</td>
<td>497</td>
<td>1</td>
<td>0.11</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>533</td>
<td>12744</td>
<td>1129516</td>
<td>6216</td>
<td>1</td>
<td>0.00</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>3</td>
<td>534</td>
<td>15707</td>
<td>1129517</td>
<td>2963</td>
<td>1</td>
<td>0.01</td>
<td></td>
</tr>
</tbody>
</table>

Allgemein gilt, dass sich bei ausreichender Duplikatanzahl und dem Einzeichnen von $(l_1, s_1), \ldots, (l_m, s_m)$ der Anstieg $\alpha_k$ mit zunehmendem $k$ verringert (s. Tabelle 7.1). Je geringer der Anstieg $\alpha_j$ ist, desto signifikanter sind die Duplikate mit Länge $l_k$ und $k \geq j$.

Die Bestimmung aller Anstiege $\alpha_k$ über $l_k, s_k$ sowie $l_{k+1}$ und $s_{k+1}$ ist mittels einer SQL-Anweisung möglich. Für einen persistenten Ergebnisbestand wurde die duplikatklassenweise Berechnung jeweils über eine summary table realisiert. Die Abb. 7.3 auf Seite 88 zeigt die Visualisierung der entsprechenden Ergebnisse aus Tabelle 7.1. Dieser Kurvenverlauf ist cha-

Abbildung 7.2: Visualisierung des Anstiegs $\alpha_k$
KAPITEL 7. STATISTISCHE & FUNKTIONELLE DUPLIKATDATENAUSWERTUNG

Abbildung 7.3: Visualisierung von Daten aus Tabelle 7.1

rakteristisch für alle anderen Abbildungen, die deswegen auch nicht abgebildet sind. Es ist sehr deutlich zu erkennen, dass anfangs der Anstieg fast 90° beträgt und erst mit großem \( l_k \) (≈ 560) bei \( s_k ≈ 1.130.000 \) allmählich abflacht.

Abb. 7.3 zeigt, dass die Duplikatdaten sehr sensitiv sind und die signifikantesten Duplikate 'rechts vom Knick' liegen. Aus Sicht der DB lassen sich diese Duplikate durch die flachen Anstiege \( \alpha_k, k ≥ k' \) ab gewissem \( k' \) identifizieren.

Die topologisch - statistische Analyse basiert folglich auf einem sehr sensitiven Datenbestand.

Durch zusätzliche Betrachtung der Duplikatarmkoordinaten ist festzustellen, dass manche Duplikatarme mit gleichen Koordinaten im unmaskierten Humangenom zu dutzenden bis hunderten verschiedener Duplikate gehören. Diese Feststellung ist nicht nur auf die Sensitivität der Duplikatdaten zurückzuführen, sondern beruht auch auf der Transitivität: Falls \( D_1, D_2 \) die Duplikatarme eines Duplikats \( D ∈ \mathcal{D} \) sind und \( E_1, E_2 \) die Duplikatarme eines Duplikatarmes \( E ∈ \mathcal{D} \) derselben Duplikatklasse mit \( D_2 = E_1 \), so ex. ein Duplikat \( F \) mit dem Duplikatarmen \( D_1 \) und \( E_2 \). Dabei sinkt die Anzahl solch mehrfach vorkommender Duplikatarme von Repeats (Palindromen) zu native Repeats (native Palindromen) deutlich.

Solche mehrfach vorkommenden Duplikatarme findet man im unmaskierten Humangenom kaum.

\[ \text{Abb. 7.3: Visualisierung von Daten aus Tabelle 7.1} \]

---

\( \text{gilt nicht für supermaximal Repeats und Tandems} \)
7.4 Beschreibung der Untersuchungsziele und Ergebnis-
    auswertung

Wie bereits in Abschnitt 3.3 auf Seite 27 ausgeführt, bestehen die Chromosomen nicht aus einer
zufälligen Basenabfolge, sondern grundsätzlich aus einer ‘wohldefinierten’ Anordnung
von (Abschnitten von) Basenabfolgen. Wenn es folglich eine Systematik beim Aufbau von
Chromosomen gibt, so stellt sich die Frage, ob und ggf. wie die Duplikate in eine Systematik
einbezogen werden.

Angenommen, die DNA-Sequenz des einen Duplikatarms ist eine ‘Sicherheitskopie’ der
DNA-Sequenz des anderen Duplikatarms. Dann ist es plausibel anzunehmen, dass (die An-
ordnung der Basen) dieser DNA-Sequenz eine Bedeutung zukommt. Die Bedeutung könnte
derin bestehen, wichtige Informationen - bspw. für die Genexpression - vor Verlust durch
Veränderung (Mutationen, Translokationen, . . .) zu bewahren. Andererseits kann es sich bei
den Duplikaten auch um Transposons handeln, die kleine DNA-Sequenzen der Größenord-
nung von ≈ 1000 Basen sind [L04]. Transposons verbreiten sich selbst innerhalb des Genoms,
ohne selbst bei der (Weiter-)Entwicklung des Organismus beizutragen. In diesem Fall wären
die Daten wohl ‘verrauscht’. Eine Untersuchung, ob Duplikat(arm)e in Genabschnitten lie-
gen, ist daher naheliegend.

Falls, Duplikat(arm)e in Genabschnitten liegen, so ist die Frage offen, ob und wenn ja wo-
durch sich diese Duplikate ggf. von denen unterscheiden, die sich möglicherweise ‘nur’ in
intergenischen Abschnitten befinden. Lassen sich sogar charakteristische Kenngrößen fest-
stellen, so hätte man einen Ansatzpunkt, um auch auf Grundlage syntaktischer und stati-
stischer Daten Gene auf vergleichbaren eukaryontischen Genomen zu vermuten.

7.4.1 Quantitativ - topologische Analyse

Die quantitativ - topologische Analyse bezieht sich auf die Beantwortung der Frage, wieviel
Prozent der Duplikate sich pro carrier_id und param_id auf die in Abb. 7.1 auf Seite 85
angegebenen Topologien (außer E) aufteilen. Dazu wurden mehrere dutzend summary ta-
bles angelegt, die auch alle Ergebnisse für die qualitativ - topologische Analyse enthalten.
Für einige maskierte Chromosomen liegen entweder keine Ergebnisse vor oder die Ergeb-
nisse beruhen auf einer sehr geringen Anzahl von Datensätzen. Diese beiden Sachverhalte
erschweren eine systematische Analyse erheblich und die Ergebnisse sind durch die geringe
Anzahl an Duplikatdatensätzen möglicherweise nicht repräsentativ. Daher muss aus Zeitnot
auf eine statistische Analyse der maskierten Humanchromosomen verzichtet werden. Die
nachfolgenden Ergebnisse beziehen sich daher ausschließlich auf das unmaskierte Human-
renom. Die Sequenzgaps beeinflussen die Ergebnisse teilweise ganz erheblich. Das einige
Chromosomen dabei überproportional lange Sequenzgaps haben, fällt weniger ins Gewicht,
vielmehr, dass manche dieser Sequenzgaps sich (fast) über einen ganzen Chromosomenarm
erstrecken (s. u.).

Stellvertretend für die anderen Duplikatklassen sind in Abb. 7.4 auf Seite 90 die Ergebnisse
der Palindrome mit param_id = 3 angegeben. Die restlichen sieben Abb. befinden sich in
Anhang D.2 auf Seite 152 ff.

Auf den ersten Blick sehen sich alle acht Abb. sehr ähnlich. Dort sind je vier Balken pro
Chromosom zu erkennen, die den prozentualen Anteil der Duplikate an der Topologie kenn-
zeichnen. Weil für das Chromosom der prozentuale Anteil jeweils bei 100% liegt, kann auf
diese Einzeichnung verzichtet werden. Die Ergebnisse auf den Chromosomen 13, 14, 15, 21
und 22 sind größteils auf ihre (fast) völlig unsequenzierten p-Chromosomenarme zurückzu-
führen. Beim Y-Chromosom ist ein Großteil des q-Chromosomenarms noch unsequenziert.
KAPITEL 7. STATISTISCHE & FUNKTIONELLE DUPLIKATDATENAUSWERTUNG

Abbildung 7.4: Proz. Aufteilung der Palindrome in diverse Topologien im unmaskierten Humangenom

Davon abgesehen liegen i. A. 50% - 60% der Duplikate auf genau einem der beiden Chromosomenarme. Wesentlich prägnanter sind die Ergebnisse für die genischen und intergenischen Abschnitte. Obwohl der prozentuale Genanteil (inklusive Introns) am Chromosom jeweils bei rund 30% - 40% liegt, sind es aber nur ca. 4% - 12% aller Duplikate, die sich in (irgend-)einem genischen Abschnitt befinden und damit unterrepräsentiert sind. Dem gegenüber stehen ca. 40% - 60% der Duplikate in (beliebigen) intergenischen Abschnitten. Die Chromosomen 17, 19 und 22 bilden eine Ausnahme, die hier nicht weiter beleuchtet wird. Für die längsten Chromosomen korreliert der Prozentwert mit dem der Chromosomenarme. I. d. R. liegen sogar nur ca. 3% - 8% der Duplikate innerhalb ein- und derselben zytogenetischen Bande, wobei die Werte für die kleinsten Chromosomen deutlich darüber liegen. Generell lassen sich zwischen Repeats und Palindromen, Repeats und native Repeats, usw., nur wenige Unterschiede ausmachen, die Ansatzpunkt für detailiertere Analysen wären. Insgesamt sind die Ergebnisse zu unspezifisch.

7.4.2 Qualitativ - topologische Analyse


Strukturell gleichen sich noch Repeats und Palindrome sowie native Repeats und native Palindrome. Jedoch enden die Gemeinsamkeiten bei der durchschnittlichen Duplikatarmlänge. Auffällig sind die langen Palindromarme im Genbereich des Y-Chromosoms. Auch bei Berücksichtigung, dass noch ca. die Hälfte (s. Tabelle C.1 auf Seite 129) des Chromosoms unsequenziert ist, sind diese Ergebnisse aussagekräftig [SK03].


Abgesehen von den zytogenetischen Banden haben sich die durchschnittlichen Duplikatarmlängen in etwa (mindestens) verdoppelt (Abb. D.4, D.5, D.8 und D.9). In den Abb. D.4 und D.8 setzen sich die Balken der zytogenetischen Banden deutlich von den anderen ab, u. a., weil nur relativ wenige (3% - 8%) Duplikate in zytogenetischen Banden liegen und diese dann überdurchschnittlich lang sind.

Abbildung 7.5: Palindromarmlängen in diversen Topologien im unmaskierten Humangenom

![Diagramm zur Darstellung der Palindromarmlängen in diversen Topologien im unmaskierten Humangenom](image-url)
7.5 Kapitelzusammenfassung

Das Fazit der statistisch-topologischen Analyse ist gleich in mehrfacher Hinsicht unvollständig. Zum einen verzerren teilweise (noch) Sequenzlücken die Ergebnisse (s. bspw. Abb. 7.4). Zum anderen scheinen die Duplikate der unmaskierten Humanchromosomen zu sensitiv, die der maskierten Humanchromosomen zu selektiv zu sein. Im ersten Fall ist ein (langfristiger) methodischer Schritt zur Behebung dieses Problems möglicherweise das in Abschnitt 6.2.2.4 auf Seite 78 angesprochene *extended native clustering*. Kurzfristig ist lediglich eine mittels SQL einfach zu realisierende Verschärfung der Auswahlkriterien (prozentuale Identität, Mindestlänge, ...) denkbar.

Als verwertbare Erkenntnisse bleibt festzuhalten, dass

- es kaum vorhandenen Unterschiede zwischen Repeats (native Repeats) und Palindromen (native Palindromen) mit Ausnahme des Y-Chromosoms gibt.
- auch kein bemerkenswerter relativer, quantitativer Unterschied zwischen den Duplikaten existiert.
- nur sehr wenige Duplikate in ein und derselben zytogenetischen Bande liegen, diese dann aber überproportional lang sind.
- Duplikate überwiegend in intergenischen Abschnitten liegen.


10 eigentlich neun Exons, davon sind das erste und letzte aber wesentlich länger als die verbliebenen sieben
Kapitel 8

Grafische Ausgabe der Duplikate

Obwohl durch zahlreiche Verarbeitungsschritte die Duplikatdaten aufbereitet wurden und dabei das native clustering ganz entscheidende Vorteile bietet, sind die Möglichkeiten zur Entdeckung ‘interessanter’ Duplikate im Rahmen des DBS überwiegend ausgereizt. Eine Beurteilung der Duplikatdaten zur Aufdeckung charakteristischer Strukturen ist (nur) durch eine Visualisierung möglich.

Die in Abb. 8.1 angegebenen zu durchlaufenen Phasen, ausgehend von einem Eingabeformular bis hinab zum Zugriff auf die DB, werden in den einzelnen Abschnitten dieses Kapitels separat erläutert.

Es mag durchaus Programme geben, die mittels weniger Mausklicks die gewünschten Duplikatdaten aus der Datenbank holen und visualisieren. Diese Programme sind aber mit hoher Wahrscheinlichkeit nicht plattformunabhängig und nur einer quantitativ kleinen Nutzergruppe vorbehalten.


Abbildung 8.1: Schema der Verarbeitungskette zur grafischen Darstellung der Duplikate
KAPITEL 8. GRAFISCHE AUSGABE DER DUPLIKATE

8.1 Aufbau und Beschreibung des Eingabeformulars

Das Eingabeformular ist in Abb. 8.2 auf Seite 95 dargestellt. Im Idealfall sollte es auf Knopfdruck mit den in der DB aktuell vorhandenen Daten der Schemata admin und sequence erstellt werden, da diese zur Navigation durch das Webformular gehören und angezeigt werden müssen (s. u.).


II repräsentiert zunächst einmal alle in SYNTENY befindlichen Sequenzklassen (Schema duplicates), die über eine Checkbox an- oder abwählbar sind. Die optionale Angabe von minimalen oder maximalen Diagonalwerten dient dem Herausschneiden, bspw. wenn eine lange und (damit) zugleich signifikante Duplikatdiagonale von anderen (zu) kurzen, weil 'verrauschten' und störenden Duplikatgeraden, flankiert wird.

Die optionale Angabe eines maximalen Abstands beider Duplikatarme (außer Tandems) ist nicht nur eine ordinäre Filtermöglichkeit, sondern reflektiert auch biologische Sachverhalte, auf die an dieser Stelle nicht näher eingegangen wird.

max. dup distance und min. cluster size sind nur bei Anwahl von III (s. u.) verfügbar. Erstgenanntes legt den maximalen Abstand zwischen zwei Duplikaten (desselben Sequenztyps) fest und letztgenanntes den minimalen Umfang eines jeden Clusters, um eingezeichnet zu werden.

Die Angaben zu den Klickboxen bezüglich vmatch id entsprechen denen der DBT von Tabelle 4.6 auf Seite 37 bis Tabelle 4.9 auf Seite 39. Zusätzlich lässt sich über 'all' die vmatch id - Beschränkung aufheben (s. a. VII).

Über arm color und arm line type sind Vorgaben zur graphischen Ausgabe der Duplikatarme vornehmbar. So unterscheiden sich zwar Palindrome von den anderen Sequenzklassen durch ihre Darstellung als Antidiagonale, jedoch nicht Repeats und Tandems. Dieses Problem lässt sich über geeignete Angaben zu arm color und arm line type umgehen.

Mit Anwahl des Einzeichnens von MST unter III werden die korrespondierenden Eingabefelder unter II für Eingaben freigeschaltet. Die beiden anderen Auswahlelemente mst color und mst line type sind analog zu den Duplikatarmen zur visuellen Differenzierung angegeben. Die Berechnung von MST übernimmt ein C - Programm, das in Abschnitt 8.7 auf Seite 105 ausführlich erläutert wird.

Die Gruppierungsboxen IV und V sind gewissenmaßen nur Platzhalter für zukünftige Funktionalitätserweiterungen des Webformulars. Das Problem der funktionalen Daten (Gene, Exons) besteht grundsätzlich in den externen Datenquellen. Jeder Anbieter (wie ENSEMBL) von funktionalen Daten kann ein eigenes Datenformat verwenden, dass noch nicht einmal datenbankkonform vorliegen muss. Ohne vorliegende Daten ist aber eine geeignete Datenintegration nicht praktikabel und damit auch kein einheitlicher Zugriff auf die funktionalen Daten möglich.

Die Schwierigkeit des Einzeichnens von unbekannten oder maskierten Sequenzabschnitten beruht auf den unterschiedlich langen Sequenzzuschnitten (einige Kbp bis gesamtes Chromosom). Unbekannte oder maskierte Sequenzabschnitte sind als ausgefüllte Trapeze einzuzeichnen, weil Duplikatdaten aufgrund einer Achsensymmetrie an der Winkelhalbierenden

1d. h. die Summe der horizontalen Ausdehnung (in Bp) und der vertikalen Ausdehnung (in Bp)
2Die vier Kanten bestehen aus einem Abschnitt der Ordinate und der Winkelhalbierenden sowie zwei zur Ordinate orthogonalen Geraden.
Abbildung 8.2: Eingabeformular zur Duplikatvisualisierung
KAPITEL 8. GRAFISCHE AUSGABE DER DUPLIKATE


VI enthält vier Eingabefelder zum Eingeben von Zoomkoordinaten in der Reihenfolge \( x_{\text{min}}, x_{\text{max}}, y_{\text{min}}, y_{\text{max}} \). Beginnend mit ganzchromosomalen Betrachtungen lassen sich sukzessive interessante Regionen visuell ermitteln.


Mit zunehmenden Eingabe- / Auswahlmöglichkeiten entstehen jedoch möglicherweise komplexere Abhängigkeiten zwischen den Elementen. Sobald sich bspw. in I die Version ändert, müssen in `Species` und `Carrier` die Daten aktualisiert werden; wird `Species` geändert, immerhin noch die Daten von `Carrier`. Auch müssen alle Eingabefelder nicht nur auf die Eingabe von Zahlen kontrolliert werden, sondern auch auf ihre Zulässigkeit. Insgesamt übernehmen mittlerweile rund 50 in Javascript codierte Funktionen diverse Kontrollmaßnahmen. Der gesamte Javascript Anteil am vollständigen Formular - Quellcode (HTML und Javascript) macht immerhin ca. 80% aus oder knapp 100 KB.

8.2 Das Common Gateway Interface und Perl


\[^3\] Analog gilt das auch für die funktionellen Positionen, da Exons (dutzende Bp) bei ganzchromosomaler Betrachtung (dutzende Mbp) viel zu klein sind.
KAPITEL 8. GRAFISCHE AUSGABE DER DUPLIKATE

8.3 Datenbankzugriff

Aus der Sicht von Perl ist es unerheblich, ob unter Einbeziehung der übermittelten Daten des Webformulars eine SQL-Anweisung abgesetzt oder eine spezielle UDF aufgerufen wird. Eine UDF weist jedoch gegenüber einer SQL-Anweisung einige Vorteile bezüglich Kapselung und Optimierung der Abfrage(n) auf, d. h.,

- eine vorherige Überprüfung der Eingabeparameter auf ihre Zulässigkeit.
- ein Ausnutzen von besonderen Sachverhalten verbessert die Laufzeit und reduziert ggf. die Ergebnismenge (s. u.).
- durch das Verbergen der 'statischen' SQL-Anweisungen in den UDF’s lassen sich optimale Indizes anlegen, die immer ausgenutzt werden (s. u.).
- die UDF’s lassen sich unter Ausnutzung aller obiger Vorteile auch in anderen Programmiersprachen / Skriptsprachen verwenden.


- Eingabeparameter von duplicates.get_pal_cl_cand:
  - carrier_id
  - param_id
  - dv_min (dv_max) - minimaler (maximaler) Diagonalwert zur Entfernung flankierender, 'versenkelter' Palindrome
  - sI,eI,sII,eII - Zoomkoordinaten; die Koordinaten des ersten (zweiten) Palindromarmes müssen zwischen sI und eI (sII und eII) liegen
  - max_dup_arm_dist - maximaler Abstand (bei Überlappung = 0) zwischen beiden Palindromarmen; korreliert teilweise mit den Zoomkoordinaten
  - max_dup_dist - maximaler Abstand zwischen zwei Palindromen, die für MST-Berechnungen in Betracht kommen (s. u.)

- Ausgabeparameter duplicates.get_pal_cl_cand:
  - s1,e1,s2,e2 - Koordinaten beider Palindromarme
  - length - Länge eines jeden Palindromarms
  - bucket_no - Wert für MST - Vorverarbeitung (s. Abschnitt 8.6 auf Seite 103)

4 gemeint ist, dass die Struktur der SQL-Anweisung statisch ist, nicht aber die Parameter
5 Akronymbildung wegen einer Längenbegrenzung von Funktions- und Prozedurnamen auf max. 18 Zeichen; bedeutet get_palindrome_cluster_candidates

97
Bei den Repeats, den native Repeats, den Palindromen und den native Palindromen sind auch Datensätze zu genau einem Chromosom (respektive carrier id) aber aber mehreren param id’s wählbar. Dadurch können sich wie beschrieben Duplikate mit unterschiedlichen param id’s überlappen, die aber über entsprechende overlap id - Attributwerte herausgefiltert werden können. Diese zusätzliche Filterung übernimmt jeweils eine eigene SQL PL UDF, bspw. duplicates.get_palov_cl_cand mit gleichen Eingabe- und Ausgabeparametern, wie oben.

Im Gegensatz zu den bisherigen (je einmal durchzuführenden) Verarbeitungsschritten der DB (Redundanzerkennung und -beseitigung; native Clustering, . . .) gibt es wenige, aber wesentliche Unterschiede: Die Anforderungen an die Datenverarbeitung müssen jetzt möglichst in Echtzeit (soft real time) durchgeführt werden, da der Nutzer nicht die Geduld aufbringen wird, minuten- oder stundenlang auf Ergebnisse zu warten. Die Auswirkungen des Echtzeit- effekts sind umso gravierender, da der Nutzer wohl mehrere Anfragen stellen wird.

Um diese Anforderungen bestmöglich umzusetzen, lassen sich folgende Vorkehrungen treffen, deren Auswirkungen am Beispiel obiger UDF dokumentiert werden:

- ‘On - the - fly’ Datenvorverarbeitung für MST - Berechnung (s. Abschnitt 8.5 auf Seite 100).
- Ggf. Anlegen von optimalen Indizes mit Hilfe von db2advis (s. Abschnitt 8.6 auf Seite 103).

Weil die Datenvorverarbeitung auf graphentheoretischen Konzepten aufbaut, ist noch eine Einführung in graphentheoretische Grundlagen bzw. Definitionen im anschließenden Abschnitt 8.4 notwendig.

### 8.4 Graphentheoretische Grundlagen

Die bildliche Darstellung von ‘benachbarten’ Duplikaten die ein Cluster bilden, wird durch einen minimal spannenden Baum (minimum spanning tree, kurz MST) realisiert.

Ein MST ist ein Graph mit besonderen Eigenschaften, die nachfolgend sukzessive herausgearbeitet werden.

**Definition 8.1 (Graph \( \mathcal{G}, V(\mathcal{G}), E(\mathcal{G}) \))** Ein Graph \( \mathcal{G} \) besteht aus Knoten (und Kanten). Die Menge aller Knoten in \( \mathcal{G} \) wird mit \( V(\mathcal{G}) \) bezeichnet. Die Menge aller Kanten in \( \mathcal{G} \) wird mit \( E(\mathcal{G}) \) bezeichnet.

Die Knoten eines Graphen \( \mathcal{G} \) repräsentieren hier die Duplikate genau einer der sechs Sequenzklassen, sodass im Kontext der Graphdefinition Knoten und Duplikate Synonyme sind, d. h.,

\[
V(\mathcal{G}) \in \{ \mathcal{P}, \mathcal{P}_N, \mathcal{R}, \mathcal{R}_N, \mathcal{S}, \mathcal{T} \}
\]

Zwei Knoten \( v_i \in V(\mathcal{G}) \) mit den Koordinaten \( A = (s_1, s_2) \), \( B = (e_1, e_2) \) und \( v_j \in V(\mathcal{G}) \) mit den Koordinaten \( C = (\tau_1, \tau_2) \), \( D = (\tau_1, \tau_2) \) werden, wie in Abb. 8.3 auf Seite 99 dargestellt, durch eine Kante \( e_{i,j} \in E(\mathcal{G}) \) mit den Koordinaten

\[
\left( \frac{s_1 + e_1}{2}, \frac{s_2 + e_2}{2} \right), \left( \frac{\tau_1 + \tau_1}{2}, \frac{\tau_2 + \tau_2}{2} \right)
\]

(8.1)

\[\text{ohne param id}\]

\[\text{in diesem Fall Repeats}\]
verbunden. Der entscheidende Vorteil dieser Festlegung der Kantenkoordinaten ist die Eindeutigkeit. Üblicherweise würde man wohl eher die Duplikatarmenden \(((A, C), (A, D), (B, C), (B, D))\) mit minimalen Abstand zueinander verbinden. In Abb. 8.3 ist der Abstand zwischen \((A, C)\) und \((B, D)\) gleich und somit eine Kanteneinzeichnung nicht deterministisch.

Abbildung 8.3: Kanteneinzeichnung im MST

Nach (8.1) existiert immer eine Kante, sie ist daher immer definiert. Durch die Redundanzbeseitigung und das native clustering gibt es keine zwei Duplikate mit gleicher \textit{carrier\_id} und gleicher \textit{param\_id} derart, dass \(\frac{s_1+e_1}{2} = \frac{s_1+e_1}{2}\) und \(\frac{s_2+e_2}{2} = \frac{s_2+e_2}{2}\) gilt, m. a. W., die Kante zu einem Punkt zusammenschrumpft. Bei Kanten mit identischer \textit{carrier\_id} aber unterschiedlicher \textit{param\_id} verhindert der \textit{overlap\_id} - Attributwert in Zusammenhang mit dem Primärschlüssel die Auswahl sich vollständig überlappender Duplikate. Deshalb kann eine Kante ebenfalls nicht punktförmig sein.

Weil eine Kante nach Definition (8.1) die Mittelpunkte zweier Duplikate verbindet, ist die biologische Interpretation einer Kante nicht eindeutig. Überbrückt die Kante z. B. in Relation zu den Duplikat(arm)längen eine längere Distanz, so repräsentiert sie i. d. R. zwei 'unähnliche' DNA - Subsequenzen.

\textbf{Definition 8.2 (Kantenlänge, }\textit{l}(e_{i,j})\text{)} \quad Sei \(\mathcal{G}\) ein Graph und \(e_{i,j} \in \mathcal{G}\) eine Kante mit den Koordinaten von (8.1). Dann ist die Kantenlänge \(\textit{l}(e_{i,j})\) definiert als

\[
\textit{l}(e_{i,j}) = \left( \left\lvert \frac{s_1+e_1}{2} - \frac{s_1+e_1}{2} \right\rvert + 1 + \left\lvert \frac{s_2+e_2}{2} - \frac{s_2+e_2}{2} \right\rvert + 1 \right)
\]

\[
= \frac{s_1+e_1 - s_1 - e_1}{2} + \frac{s_2+e_2 - s_2 - e_2}{2} + 2
\]

\[
= \frac{|s_1+e_1 - s_1 - e_1| + |s_2+e_2 - s_2 - e_2| + 4}{2}
\]

Weil sich den Kanten i. A. kein Richtungssinn analog zur Leserichtung der Duplikate zuorden lässt, liegt ein \textit{ungerichteter} Graph \(\mathcal{G}\) vor.

Die beiden folgenden Definitionen sind angelehnt an [T96].
KAPITEL 8. GRAFISCHE AUSGABE DER DUPLIKATE

Definition 8.3 (Kantenzug, Weg) Eine Folge von Kanten \( e_{i,j} \in E(G) \) eines Graphen \( G \) heißt Kantenzug, gdw. es eine Folge von Knoten \( v_1 \in V(G), \ldots, v_k \in V(G) \) mit \( k \in \mathbb{N}^+, 2 \leq k \) gibt, die folgende Bedingung erfüllt:

\[
\forall i \in \mathbb{N}^+, 2 \leq i < k : e_{i,i+1} \in E(G)
\]

Ein Kantenzug eines Graphen \( G \) heißt Weg, gdw. alle verwendeten Kanten verschieden sind, d. h.,

\[
\forall i \in \mathbb{N}^+, j \in \mathbb{N}^+, 2 \leq i < j < k : \nexists v_{i,i+1} \in V(G), v_{j,j+1} \in V(G) : v_{i,i+1} = v_{j,j+1}.
\]

Nach Def. 8.3 lässt sich dann ein MST wie folgt definieren:

Definition 8.4 (Baum, MST) Ein ungerichteter Graph \( G \) heißt Baum, gdw. es entweder zwischen zwei beliebigen Knoten \( v_i \in V(G), v_j \in V(G) \) genau einen Weg gibt oder \( G \) aus genau einem Knoten besteht, d. h., \( |V(G)| = 1 \).

\( G \) heißt minimal spannender Baum (MST), gdw. \( G \) ein Baum ist und die Summe der Kantellängen minimal (= \( l_{min} \)) ist, d. h.,

\[
\exists l < l_{min} : \sum_{e_{i,j} \in E(G)} l(e_{i,j}) = l.
\]

Ein MST mit \( |V(G)| \) Knoten hat \( |E(G)| = |V(G)| - 1 \) Kanten.

Nach Definition 8.4 muss ein MST nicht eindeutig sein. Nach den Algorithmen von PRIM und KRUSKAL (s. bspw. [196]) werden MST’s effizient berechnet. Die Laufzeit bei Standardimplementierung beträgt jeweils \( \Theta(|V(G)| \log |E(G)|) \). Ein Vergleich der beiden vorgestellten Algorithmen ist schwierig, da diese sehr stark von der gewählten Datenstruktur abhängen ([196]).

Der hier implementierte Algorithmus von KRUSKAL soll nur kurz skizziert werden, um daran die Methoden der Datenvorverarbeitung zu erläutern. Eine genauere Beschreibung findet sich z. B. in [196].

Der Algorithmus von KRUSKAL geht von einem Graphen \( G \) aus, der nur Knoten enthält, d. h., \( E(G) = \emptyset \). Alle \( \frac{|V(G)||V(G)|-1}{2} \) möglichen Kanten von \( G \) werden nach aufsteigender Kantellänge sortiert. Zwei Knoten \( v_i \in V(G), v_j \in V(G) \) werden durch eine Kante \( e_{i,j} \in E(G) \) miteinander verbunden, gdw. es (noch) keinen Weg zwischen \( v_i \) und \( v_j \) gibt. Nach Definition 8.4 ist \( G \) nach dem Einfügen von \( |V(G)| - 1 \) Kanten ein MST.

Bei den jeweils rund 1,8 Mio. Repeats bzw. Palindromen des unmaskierten humanen X-Chromosoms und \( max_{dup\_dist} = \) Chromosomenlänge existieren jeweils ca. \( 1,62 \cdot 10^{12} \) mögliche und nach Kantellänge zu sortierende Kanten, obwohl der MST letztendlich eine Kante weniger als Knoten hat. Die Bestimmung eines solchen MST ist heutzutage im wahrsten Sinne des Wortes (noch) praktisch unberechenbar. Bei (biologisch) ’vernünftiger’ Wahl von \( num_{dup\_dist} \) und der Datenvorverarbeitung lässt sich der Berechnungsaufwand drastisch reduzieren.

8.5 Datenvorverarbeitung

Der erste Schritt zur Laufzeitverbesserung besteht in einer effizienteren Bestimmung der für den KRUSKAL - Algorithmus in Frage kommenden Kanten. In Abhängigkeit von \( max_{dup\_dist} \)
KAPITEL 8. GRAFISCHE AUSGABE DER DUPLIKATE

sind nur all jene Duplikate für den KRUSKAL-Algorithmus relevant, deren Distanz \( \leq \text{max}_\text{dup}\text{dist} \) ist. Die Duplikatdistanz (in Bp) entspricht aber nicht der Kantenlänge. Die Distanz zwischen zwei zweidimensionalen Punkten \( p \) und \( q \) wird durch die Manhattandistanz \( d \) (ebensfalls in Bp) definiert:

**Definition 8.5 (Manhattandistanz, \( d \))** Seien \( P = (x_1, y_1) \) und \( Q = (x_2, y_2) \) Punkte mit den entsprechenden Koordinaten. Dann ist der Abstand \( d \) zwischen \( P \) und \( Q \) definiert als

\[
d(P, Q) = \max\{0, (|x_1 - x_2| - 1) + (|y_1 - y_2| - 1)\}.
\]

Da die Knoten alias Duplikate eines Graphen jedoch diagonale Strecken sind, deren Koordinaten jeweils durch zwei Punkte repräsentiert werden, ist noch eine exakte Distanzfestlegung zwischen zwei Duplikaten durch Definition 8.6 notwendig:

**Definition 8.6 (Duplikatdistanz)** Seien \( A \in \mathcal{D} \) und \( B \in \mathcal{D} \) gegeben. Seien ferner \( P_1, P_2 \) bzw. \( Q_1, Q_2 \) die Koordinaten der beiden Duplikat(arm)e. Dann ist die Duplikatdistanz \( D(A, B) \) definiert als

\[
D(A, B) = \min\{d(P_1, Q_1), d(P_1, Q_2), d(P_2, Q_1), d(P_2, Q_2)\}
\]

Die Laufzeitverbesserung zur Kantenbestimmung wird durch geschicktes Gruppieren der Duplikate über einen Hash erreicht. Das Gruppieren erfolgt unter Einbeziehung der längsten Duplikat(arm)länge \( l_{\text{max}} \). Die Duplikate fallen nach Gleichung (8.2) in den Bucket \( b[i] \).

\[
i = \left\lfloor \frac{s_1}{l_{\text{max}} + \text{max}_\text{dup}\text{dist} + 1} \right\rfloor
\]

mit \( s_1 \) als Startposition des ersten Duplikatarmes.

**Satz 8.1** Sei \( G \) ein Graph. Seien \( D_i \in V(G) \) mit den Duplikatarmen \( \overline{D}_i(\tilde{s}_i, \tilde{e}_i) \) sowie o. B. d. A. \( \tilde{s}_i < \tilde{s}_j \), \( D_j \in V(G) \) mit den Duplikatarmen \( \overline{D}_j(\tilde{s}_j, \tilde{e}_j) \) sowie o. B. d. A. \( \tilde{s}_j < \tilde{s}_j \) jeweils Knoten aus den Buckets \( b[i] \) und \( b[j] \) mit \( i \in \mathbb{N}_0, j \in \mathbb{N}_0 \) und o. B. d. A. \( i + 2 \leq j \). Dann gilt:

1. \( \tilde{s}_i < \tilde{s}_j \), d. h. die Startposition \( \tilde{s}_i \) von \( \overline{D}_i \) ist kleiner als die Startposition \( \tilde{s}_j \) von \( \overline{D}_j \).
2. \( \forall D_i, D_j : D(D_i, D_j) > \text{max}_\text{dup}\text{dist} \)

**Beweis:**
Sei \( z := (l_{\text{max}} + \text{max}_\text{dup}\text{dist} + 1) \Rightarrow z > 0. \)

1. Durch Umstellung von Gleichung (8.2) gilt:

\[
\tilde{s}_i = i \cdot z + k_i, \quad k_i \in \mathbb{N}_0, 0 \leq k_i < z
\]

\[
i = \frac{\tilde{s}_i - k_i}{z}
\]

bzw.
KAPITEL 8. GRAFISCHE AUSGABE DER DUPLIKATE

\[ \bar{s}_j = j \cdot z + k_j, \quad k_j \in \mathbb{N}_0, 0 \leq k_j < z \]

\[ j = \frac{\bar{s}_j - k_j}{z} \]

Substitution von \( i \) und \( j \) in \( i + 2 \leq j \) ergibt:

\[ \frac{\bar{s}_i - k_i}{z} + 2 \leq \frac{\bar{s}_j - k_j}{z} \]

\[ \bar{s}_i + \frac{(k_j - k_i) + 2z}{z} < \bar{s}_j \]

\[ -z < k_j - k_i < z \]

\[ \bar{s}_i < \bar{s}_j \]

2.

Sei \( l_i := \tau_i - \tau_i + 1 \) die Duplikatarmlänge von \( D_i \). Jeder Bucket \( b[k] \) überstreicht das Intervall \([k \cdot z, (k + 1) \cdot z - 1]\), d. h., \( \tau_i \in [i \cdot z, (i + 1) \cdot z - 1] \) und \( \tau_j \in [j \cdot z, (j + 1) \cdot z - 1] \). Den geringsten Duplikat(arm)abstand haben \( D_i \) und \( D_j \) mit \( j = i + 2 \), sowie \( \tau_i = (i + 1) \cdot z - 1 \) und \( \tau_j = (i + 2) \cdot z \). Wegen \( \tau_j > \tau_i \) liegen \( \bar{s}_j \) und \( \bar{s}_i \) um

\[ (i + 2) \cdot z - ((i + 1) \cdot z - 1) - 1 = z \]

\[ = l_{\text{max}} + \text{max}_\text{dup}_\text{dist} + 1 \]

Bp auseinander. Da \( l_i \leq l_{\text{max}} \) liegen \( D_i \) und \( D_j \) mindestens \( \text{max}_\text{dup}_\text{dist} + 1 \) Bp auseinander. Folglich gilt:

\[ D(D_i, D_j) > \text{max}_\text{dup}_\text{dist} \]

Nach Gleichung (8.2) liegt \( i \) im Bereich von null bis ca. der Chromosomenlänge. Bei Betrachtungen am Chromosomenende sind vielleicht nur einige dutzend Buckets mit Duplikaten gefüllt, aber \( i \gg 10^6 \). Wenn das Hash über ein Array implementiert wird, so muss auch für alle unbelegten Buckets Speicher alloziert werden, was sehr verschwendert sein kann. Sei \( b[i] \) der erste Bucket mit einem Duplikat. Durch die Indexverschiebung \( b[j] := b[j - i] \) mit \( j \geq i \) entfällt auf jeden Fall ein Duplikat. Die Berechnung von \( j - i \) kann vor der Eintragung durchgeführt werden, sodass doppelte Bucketeintragungen vermeidbar sind. Trotzdem ist es möglich, dass der Hash dann immer noch viele Buckets besitzt. Wollte man das Hashing auch noch auf den zweiten Duplikatarm ausdehnen, so müsste jeder Bucket \( b[i] \) selbst ein Array, \( c[k] \), sein. In Bezug auf die Anzahl der Buckets wächst damit der Speicher verbrauch in etwa quadratisch. Außerdem sollte im Normalfall ein Bucket \( b[i] \) nicht (sehr) viele Duplikate beinhalten. Diese würden sich dann mit wachsendem \( i \) auf \( c[k] \) immer ungleichmäßig verteilen, weil \( i \) von der Startposition des ersten Duplikatarmes abhängt und diese stets kleiner ist als die des zweiten Duplikatarmes. Für kleinere \( i \) ist wiederum anzunehmen, dass hier die Duplikate in den Buckets \( c[k] \) recht gleichmäßig verteilt sind, und dennoch \( c[k] \) vollständig abgehandelt werden muss, was sich negativ auf die Laufzeit auswirken sollte. Aus diesen Darlegungen werden die Duplikate eines jeden Buckets \( b[i] \) ungeordnet über

102
eine verkettete Liste verwaltet. \( b[i] \) kann (im Prinzip) keine verkettete Liste sein, weil die Duplikate in konstanter Zeit in den entsprechenden Bucket \( b[i] \) eingefügt werden müssen. Nach Satz 8.1 können alle Duplikate von \( b[i] \) nur mit den Duplikaten aus den Buckets \( b[i - 1], b[i] \) und \( b[i + 1] \) mit \( i \in N^+ \) eine Duplikatdistanz \( \leq max_{dup\_dist} \) und damit ggf. auch eine Kante in einem MST besitzen.

Für Laufzeitbetrachtungen zur Bestimmung all dieser Kantenkandidaten sei \( n \in N^+ \) die Anzahl der Duplikate und \( b \in N^+ \) die Anzahl der Buckets (\( b[0], \ldots, b[b - 1] \)). Sei \( m_i \) die Anzahl der Duplikate in \( b[i] \). Dann gilt:

\[
n = \sum_{i=0}^{b} m_i.
\]

Geht man von unsortierten Duplikaten aus, so sind ohne Hashing \( n(n-1) \) Duplikatdistanzvergleiche mit Laufzeit \( \Theta(n^2) \) notwendig. Aus Vereinfachungsgründen seien die Duplikate in den Buckets in etwa gleichmäßig verteilt, d. h.,

\[
m_i \approx \frac{n}{b}.
\]

Unter Vernachlässigung des Falles \( b[0] \) wird die Anzahl der Vergleiche im Hash ermittelt. Es ist zu beachten, dass beginnend bei \( i = 0 \) nur die Buckets \( b[i] \) und \( b[i+1] \) verglichen werden müssen, denn im folgenden Schritt, also mit \( i' := i+1 \) wurden \( b[i'-1] = b[i] \) und \( b[i'] = b[i + 1] \) bereits verglichen.

\[
\begin{align*}
\sum_{i=0}^{b-2} m_i \cdot (m_i - 1) & \div 2 + m_i \cdot m_{i+1} \\
= & \sum_{i=0}^{b-2} \frac{m_i^2 - m_i}{2} + m_i^2 \\
= & \sum_{i=0}^{b-2} \frac{3 \cdot m_i^2 - m_i}{2} \\
= & \frac{3}{2} \left( \frac{n}{b} \right) ^2 - \frac{1}{2} \left( \frac{n}{b} \right) \\
\approx & \frac{3}{2} \cdot n^2 \div b - \frac{1}{2} \cdot n
\end{align*}
\]

Da \( 1 \leq b \leq n \) ist die Laufzeit im schlimmsten Fall auch \( \Theta(n^2) \). Nach Gleichung (8.2) wächst \( b \) mit abnehmenden \( max_{dup\_dist} \), was auch biologisch Sinn ergibt. In der Praxis sollte die Laufzeit mit Hashing aber spürbar schneller sein als ohne.

### 8.6 Umsetzung im Datenbanksystem

Die in Abschnitt 8.3 auf Seite 97 beispielhaft vorgestellte UDF \( duplicates.get\_pal\_cl\_cand \) zur Filterung irrelevanter Palindrome nimmt durch Hinzufügen von zwei neuen SQL-Anweisungen und der Modifikation der bisherigen SQL-Anweisung für jedes Palindrom die Bucketberechnung, d. h., die Berechnung von \( i \) vor.

\(^{*}\)nur relevant bei kleinem \( b \)
Abfrage 8.1 SET MAX_DUP_LEN = (SELECT MAX(LENGTH) FROM Duplicates.Palindromes WHERE CARRIER_ID = cid AND PARAM_ID = pid AND DIA_VALUE BETWEEN dv_min AND dv_max AND STARTPOS1 >= sI AND ENDPOS1 <= eI AND STARTPOS2 >= sII AND ENDPOS2 <= eII);

Abfrage 8.2 SET MIN_BUCKET_NO = (SELECT MIN(STARTPOS1 / (MAX_DUP_LEN + max_dup_dist + 1))) FROM Duplicates.Palindromes WHERE CARRIER_ID = cid AND PARAM_ID = pid AND DIA_VALUE BETWEEN dv_min AND dv_max AND STARTPOS1 >= sI AND ENDPOS1 <= eI AND STARTPOS2 >= sII AND ENDPOS2 <= eII);

Abfrage 8.3 RETURN SELECT STARTPOS1, ENDPOS1, STARTPOS2, ENDPOS2, LENGTH, STARTPOS2 - (ENDPOS1 + 1), CEIL(STARTPOS1 / (MAX_DUP_LEN + max_dup_dist + 1)) - MIN_BUCKET_NO AS BUCKET_NO FROM Duplicates.Palindromes WHERE CARRIER_ID = cid AND PARAM_ID = pid AND DIA_VALUE BETWEEN dv_min AND dv_max AND STARTPOS1 >= sI AND ENDPOS1 <= eI AND STARTPOS2 >= sII AND ENDPOS2 <= eII ORDER BY STARTPOS1;

Die Abfrage 8.3 gibt für jedes betreffende Palindrom neben den vier Palindromarmkoordinaten und der Länge noch den Palindromarmabstand (STARTPOS2 - (ENDPOS1 + 1)) und eben i über CEIL(STARTPOS1 / (MAX_DUP_LEN + max_dup_dist + 1)) - MIN_BUCKET_NO zurück. Durch Abzug von MIN_BUCKET_NO aus Abfrage 8.2 wird sichergestellt, dass der erste Bucket b[0] immer belegt ist. Die sortierte Ausgabe (ORDER BY STARTPOS1) ermöglicht die Erstellung der sortierten Duplikatliste (s. Abschnitt 8.7 auf Seite 105) durch einfaches Anfügen neuer Palindromdaten an das Listenende.

Da diese drei SQL-Anweisungen entscheidend zur Gesamtauflaufzeit, d. h., inklusive Laufzeit des KRUSKAL-Algorithmus beitragen und ihre Gesamtauflaufzeit möglichst soft real time Anforderungen erfüllen sollte, ist eine Überprüfung der vorhandenen Indizes auf Optimalität erforderlich. Wie üblich wurde jeweils mit db2advis die Optimalität der vorhandenen Indizes bezüglich der SQL-Anweisung überprüft. Die Abb. 8.4 auf Seite 106, 8.6 auf Seite 107 und 8.8 auf Seite 108 zeigen die Zugriffspläne auf die Daten durch die verfügbaren Indizes samt aufgeschlüsselter timeron - Angaben an.

Die in Tabelle 8.1 auf Seite 105 angegebenen aktuellen timerons sind quasi die durch db2advis bestimmten momentanen Zugriffskosten. Findet db2advis einen besseren Index, kalkuliert es die anfallenden Zugriffskosten auf Grundlage des vorgeschlagenen Index. Das die tatsächlichen timerons von den geschätzten timerons derart abweichen, ist wohl auf das MDC zurückzuführen. Ratsam ist, den von db2advis angegebenen runstats - Befehl modifiziert
Tabelle 8.1: Kenndaten zur Indexerstellung

<table>
<thead>
<tr>
<th>Abfrage</th>
<th>aktuelle timerons nach Indexerstellung</th>
<th>geschätzte timerons nach Indexerstellung</th>
<th>tatsächliche timerons nach Indexerstellung</th>
<th>rechner. Verbeserungsfaktor</th>
<th>Anz. benötigter Indexpages</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.1</td>
<td>7418,5483</td>
<td>819,4172</td>
<td>155,2667</td>
<td>≈ 48</td>
<td>≈ 400000</td>
</tr>
<tr>
<td>8.2</td>
<td>7407,2695</td>
<td>3961,8955</td>
<td>351,3263</td>
<td>≈ 19</td>
<td>≈ 350000</td>
</tr>
<tr>
<td>8.3</td>
<td>7406,7090</td>
<td>4992,4189</td>
<td>456,2472</td>
<td>≈ 16</td>
<td>≈ 450000</td>
</tr>
</tbody>
</table>

und manuell auszuführen. Die Indexerzeugung mit runstats verbessert den (rechnerischen) Datenzugriff bis auf Faktor 48!

Anhand der benötigten Indexpages pro Query ist zu erkennen, dass pro Index - Attribut ca. 50000 Indexpages belegt werden. Allein für die Palindrome werden 4096 Bytes Seiten (400000 + 350000 + 450000) Seiten ≈ 4,9 GB an Festplattenkapazität verbraucht. Für die Indizes der Repeats wird dann in etwa der gleiche Festplattenplatz beansprucht, für die native Palindrome und die native Repeats schätzungsweise rund die Hälfte. Dementsprechend dauert das Anlegen obigen drei Indizes allein schon mehrere Stunden.

Durch die Möglichkeit des einfachen Aufrufs bspw. von duplicates.get_pal_cl_cand in externen Applikationen wird sichergestellt, dass immer ein optimaler Datenzugriff erfolgt. In den Abb. 8.5 auf Seite 106, 8.7 auf Seite 107 und 8.9 auf Seite 108 sieht man den verbesserten Datenzugriff deutlich. 'Insbesondere' wenn die Duplikatdaten und die Indexdaten bereits in den Pufferpools (also im Hauptspeicher) vorliegen, sollten die sortierten Daten der einzelnen UDF’s in nur linearer Zeit zurückgegeben werden. Liegen sie hingegen noch auf Festplatte, so beeinträchtigen aber u. U. Positionierungen des Schreib-/Lesekopfes der Festplatte den Datenzugriff auf unzusammenhängende Sektoren der Festplatte, die Laufzeit.

8.7 Clusterprogramm zur Berechnung minimal spannender Bäume

Die Aufgabe des Clusterprogramms ist es, mittels der relevanten Duplikatdaten und Parametervorgaben zu prüfen, ob sich mehrere Duplikate zu einem MST vereinen lassen und dies ggf. durch Einzeichnen einer Kante visuell zu symbolisieren. Weil es auf die Duplikatdaten über einige UDF’s wie duplicates.get_pal_cl_cand zugreift, müssen die Parameter explizit übergeben werden. Der schematische Programmablauf des Clusterprogramms ist in Abb. 8.10 auf Seite 110 angegeben.


Die anschließende Duplikatdistanzberechnung samt Laufzeit wurde im vorigen Abschnitt 8.6 auf Seite 103 dargestellt. Gilt für zwei Duplikate $D_i \in V(G)$ und $D_j \in V(G)$ $D(D_i, D_j) \leq max_{dup_{dist}}$, so werden diese ähnlich wie in Abb. 8.11 dargestellt über eine weitere Zeiger-
KAPITEL 8. GRAFISCHE AUSGABE DER DUPLIKATE

Abbildung 8.4: Zugriffsplan vor Indexerstellung von 8.1 mit timerons

Abbildung 8.5: Zugriffsplan nach Indexerstellung von 8.1 mit timerons

106
KAPITEL 8. GRAFISCHE AUSGABE DER DUPLIKATE

Abbildung 8.6: Zugriffsplan vor Indexerstellung von 8.2 mit timerons

Abbildung 8.7: Zugriffsplan nach Indexerstellung von 8.2 mit timerons
KAPITEL 8. GRAFISCHE AUSGABE DER DUPLIKATE

Abbildung 8.8: Zugriffsplan vor Indexerstellung von 8.3 mit timerons

Abbildung 8.9: Zugriffsplan nach Indexerstellung von 8.3 mit timerons
KAPITEL 8. GRAFISCHE AUSGABE DER DUPLIKATE

<table>
<thead>
<tr>
<th>Duplikatdatenzugriff über UDF’s</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verwalten der Duplikatdaten in verketteter Liste</td>
</tr>
<tr>
<td>Einfügen der Duplikatdaten in Buckets zur Duplikatdistanzberechnung</td>
</tr>
<tr>
<td>MST - Kantenkandidatenverwaltung in weiterer verketteter Liste</td>
</tr>
<tr>
<td>Sortieren nach zunehmenden Duplikatdistanzen mittels mergesort</td>
</tr>
<tr>
<td>MST - Berechnung nach KRUSKAL - Algorithmus</td>
</tr>
<tr>
<td>Entfernen zu kleiner MST’s</td>
</tr>
</tbody>
</table>

Abbildung 8.10: Schema der Verarbeitungskette des Clusterprogramms

strukturen und einer Duplikatdistanzeintragung in einer zweiten verketteten Liste, der Kantenkandidatenliste, als Kantenkandidaten eines MST vermerkt.

Die Frage, ob sich die Kantenkandidatenberechnung nicht vorteilhafter durch eine SQL-Anweisung oder eine SQL PL UDF realisieren lässt, muss verneint werden. Über SQL wird die Duplikatdistanz über einen ineffizienten Join berechnet, der sich mittels Indexüberprüfung durch db2advis nicht umgehen lässt. Über SQL PL ließe sich zwar die Duplikatdistanzberechnung über eine UDF effizienter codieren, jedoch scheitert die Datenrückgabe an der RETURN-Anweisung, die nicht in SQL PL command blocks vorkommen darf [BW03].

D. h. bspw., dass ein

```java
IF ... 
THEN RETURN ... 
ELSE RETURN ... 
END IF;
```

unzulässig ist.

Das Sortieren der Kantenkandidatenliste nach aufsteigender Duplikatdistanz übernimmt eine rekursive mergesort Funktion. Dazu werden die Elemente der Kantenkandidatenliste bezüglich der Elementanzahl (≈ n) in linearer Zeit in ein eindimensionales Array eingetragen. Gegenüber dem weit verbreiteten quicksort hat mergesort den entscheidenden Vorteil einer garantierten Laufzeit von Θ(n log n) mit n = Elementanzahl, während quicksort im schlimmsten Fall Laufzeit Θ(n²) aufweist. Sind verhältnismäßig viele Kanten (≈ \( \frac{n^2}{2} \)) zu sortieren, so kann die Gesamtlaufzeit dadurch beschleunigt werden, dass mergesort phasenweise sortiert, d. h., in etwa jeweils \( n − 1 \) Kanten sind und diese dem KRUSKAL - Algorithmus übergeben. Im Idealfall sind diese auch alle Kanten des MST, Falls nicht, so werden erneut in etwa \( n − 1 \) Kanten sortiert und dem KRUSKAL - Algorithmus übergeben. Dieser Vorgang wiederholt sich so lange, bis die \( n − 1 \) Kanten des MST gefunden sind. U. U. gilt dann für \( k \) Wiederholungen in Bezug auf den Sortieraufwand: \( k \cdot (n − 1) \ll \frac{n^2}{2} \). Diese Variante ist aber nicht implementiert worden, u. a. weil dann der Algorithmus von PRIM [T96] besser geeignet ist.

Wie bereits in Abschnitt 6.2.2.1 auf Seite 66 erläutert, ist die Rekursion u. U. kritisch und

alle verbliebenen Kanten sind mindestens so lang wie diese Kanten
mergesort zu einem späteren Zeitpunkt besser iterativ zu implementieren. Für das Erreichen von $\Theta(n \log m)$ des KRUSKAL-Algorithmus ist das Zusammenfassen der Zusammenhangskomponenten entscheidend.


Um im C-Programm die Laufzeit $\Theta(n \log m)$ zu erreichen, sind - ohne zu sehr ins Detail gehen zu wollen - einige Arrays zwecks konstantem Elementzugriffs mit unterschiedlichen Datentypen anzulegen. Ein Array $a[i]$ enthält z. B. die Anzahl der Duplikate der Zusammenhangskomponente $CC_i$, ergo $a[i] = |CC_i|$ für die innere IF-Anweisung in Abb. 8.12 auf Seite 111. Ein weiteres Array $b[j]$ verwaltet die ID’s, auf die bestimmte Zeiger der Duplikatdatenstruktur verweisen und mithelfen, die FORALL-Schleifen (loops) zu realisieren. Ein drittes Array $c[k]$ ist ein Hilfsarray, dass die Duplikate von $CC_k$ effizient über entsprechende Zeiger der Duplikatdatenstruktur verbindet, somit jeweils eine verkettete Duplikatclustelliste erzeugt und nach der Beendigung der WHILE-Schleife alle Duplikate der $CC_k$, alias eines MST, enthält.

Das Verketten von Duplikaten, die zum gleichen MST gehören, ermöglicht die Berechnung in $\Theta(m)$. Die MST - Ausdehnung $mst\_span$ wird auf Basis der Duplikatkoordinaten des MST nach der Formel

\[
(1 + \max(\text{endpos}_1) - \min(\text{startpos}_1)) + (1 + \max(\text{endpos}_2) - \min(\text{startpos}_2)) \quad (8.3)
\]


Weil die Knoten des MST keine Punkte, sondern diagonale Strecken (Duplikate) sind, können diese selbst einen kantenlosen MST bilden, der nur aus dem einen Knoten besteht. Um

\[10\] mit den Rechteckkoordinaten $((\min(\text{startpos}_2), \min(\text{startpos}_1)), (\min(\text{startpos}_2), \max(\text{endpos}_1)), (\max(\text{endpos}_2), \max(\text{endpos}_1)), (\max(\text{endpos}_2), \min(\text{startpos}_1)))$
KAPITEL 8. GRAFISCHE AUSGABE DER DUPLIKATE

01: #Init:
02: \( D_i = CC_i \) mit \( D_i(ID) = i \);
03: \( \text{num}_\text{mst}_\text{edges} = 0 \);  # Anzahl gefundener MST - Kanten
04:
05: #Loop:
06: WHILE \( (\text{num}_\text{mst}_\text{edges} \leq n - 1) \) DO
07: \{ 
08: \( e_{i,j} = \text{get}_\text{next}_\text{sorted}_\text{edge}() \);
09: IF \( (D_i(ID) \neq D_j(ID)) \) THEN  # MST - Kante gefunden
10: \{ 
12: \( \text{num}_\text{mst}_\text{edges} = \text{num}_\text{mst}_\text{edges} + 1 \);  # \( CC_i \) hat weniger Duplikate
13: IF \( (|CC_i| < |CC_j|) \) THEN  # aktualisiere kleinere CC
14: \{ 
15: \( D_i(ID) = j \);
16: \} 
19: \} 
20: \} 
21: ELSE  # \( CC_j \) hat weniger Duplikate
22: \{ 
23: \( D_j(ID) = i \);
24: \} 
26: \} 
27: \} 
28: \}

Abbildung 8.12: Schema der Duplikatliste

solche und ähnlich irrelevante, weil (zu) kleine Cluster auszublenden, müssen diese über den vorgegebenen minimalen Clusterspann \( \text{min}_\text{mst}_\text{span} \) liegen.

8.8 Grafikerstellung mit gnuplot

Um z. B. die Daten von Abb. 8.13 auf Seite 112 zu visualisieren, reicht es, wenn deren Koordinaten in einer Textdatei stehen. Über ein Gnuplot - Skript und dem (Unix- /Linux-) Aufruf

\[ \#\text{gnuplot} < \text{gnuplot}_\text{script}\.sh \]

erfolgt die Grafikerstellung in einem der vielen unterstützten Grafikformate (hier png). Da dieser Aufruf auch über das eingangs erwähnte Perl - Skript absetzbar ist, kann das Perl - Skript auch gleich die Grafik in eine HTML - Seite einbinden und zurückliefern.

Der Einsatz von Gnuplot ist verzichtbar, wenn bspw. das Perl - Skript die Visualisierung SVG\(^\text{11}\) basiert vornimmt. Dann ließen sich wegen der Interaktivität von SVG sogar noch Zusatzinformationen wie Duplikat(arm)längen usw. in der Grafik integrieren und per Mausklick

\(^\text{11}\)scalable vector graphics
KAPITEL 8. GRAFISCHE AUSGABE DER DUPLIKATE

Abbildung 8.13: Grober Auszug vom humanen Chromosom Y

abrufen. Wegen unzureichenden SVG-Kenntnissen und Zeitnot bleibt eine Grafikumstellung auf SVG ggf. eine zukünftige Option. Für ‘richtige’ Interaktivität bietet sich dann aber eher eine Realisierung durch OpenGL an.

Die Abb. 8.13 zeigt in einem groben Ausschnitt vom humanen Y Chromosom mit allen in diesem Bereich vorkommenden Palindromen und native Palindromen an. Die Abszissen-

Abbildung 8.14: Verfeinerter Auszug von Abb. 8.13
KAPITEL 8. GRAFISCHE AUSGABE DER DUPLIKATE


\(^{12}\)ENSEMBL Version 23

113
Abbildung 8.15: Verfeinerner Auszug der unteren Diagonale von Abb. 8.14
Abbildung 8.16: Verfeinerter Auszug der oberen Diagonale von Abb. 8.14
Kapitel 9

Zusammenfassung und Ausblick


Anhang A

Verwendete Abkürzungen

o. B. d. A. ohne Beschränkung der Allgemeinheit
i. d. R. in der Regel
i. e. S. im engeren Sinne
i. w. S. im weiteren Sinne
i. A. im Allgemeinen
I/O Input / Output
Abb. Abbildung
B Byte
Bp Basenpaare
c.a. circa
CLOB character large object
Bp Basenpaare
bspw. beispielsweise
bzw. beziehungsweise
d. h. das heisst
dt. deutsch
DB Datenbank
DBS Datenbanksystem
DBT Datenbanktabelle(n)
DMS database managed storage
DNA Desoxyribonukleinsäure
Def Definition
ff folgende
GB Gigabyte
gdw genau dann wenn
ggf. gegebenenfalls
HSP high - scoring segment pair
KB Kilobyte
Kbp Kilobasenpaare
LINE long interspersed elements
LOB large object
m. a. W. mit anderen Worten
max. maximal
MB Megabyte
Mbp Megabasenpaare
<table>
<thead>
<tr>
<th>Abkürzung</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>min.</td>
<td>minimal</td>
</tr>
<tr>
<td>MDC</td>
<td>multidimensional clustering</td>
</tr>
<tr>
<td>MST</td>
<td>minimum spanning tree</td>
</tr>
<tr>
<td>mum</td>
<td>maximum unique matches</td>
</tr>
<tr>
<td>o. ä.</td>
<td>oder ähnlichen</td>
</tr>
<tr>
<td>OLAP</td>
<td>online analytical processing</td>
</tr>
<tr>
<td>OLTP</td>
<td>online transaction processing</td>
</tr>
<tr>
<td>PL</td>
<td>Procedural Language</td>
</tr>
<tr>
<td>RNA</td>
<td>Ribonukleinsäure</td>
</tr>
<tr>
<td>s.</td>
<td>siehe</td>
</tr>
<tr>
<td>s. o.</td>
<td>siehe oben</td>
</tr>
<tr>
<td>s. u.</td>
<td>siehe unten</td>
</tr>
<tr>
<td>SINE</td>
<td>short interspersed elements</td>
</tr>
<tr>
<td>SMS</td>
<td>system managed storage</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>u. a.</td>
<td>unter anderem</td>
</tr>
<tr>
<td>u. U.</td>
<td>unter Umständen</td>
</tr>
<tr>
<td>UDF</td>
<td>User defined function</td>
</tr>
<tr>
<td>V.</td>
<td>Version</td>
</tr>
<tr>
<td>Vgl</td>
<td>Vergleich</td>
</tr>
<tr>
<td>z. B.</td>
<td>zum Beispiel</td>
</tr>
</tbody>
</table>
Literaturverzeichnis


[HF03] HILER, L. W.; FULTON, R. S.; et. al.: The DNA sequence of human chromosome 7.


[I01] INTERNATIONAL HUMAN GENOME SEQUENCING CONSORTIUM: Initial sequencing and analysis of the human genome.


[IBM_IV] IBM Corporation: SQL Reference Volume I.


Center of Bioinformatics, University of Hamburg, (2003)

[L04] LEWIN, B.: Genes VIII.


   Diplomarbeit, (2002)


Anhang B

Tabellen mit Laufzeiten

Als Rechner stand eine Multiprozessor - SUN - Maschine mit mehreren GB RAM und SunOS 5.8 zur Verfügung.

B.1 BLAST (ENSEMBL - VERSION 18_34)

Die BLAST - Vorverarbeitung ist bezüglich Laufzeit und jeweils wenigen 100 MB Hauptspeicherverbrauch unkritisch. Die Erstellung aller Vorverarbeitungsdateien dauerte ca. 0,5h (1213,2s + 553,4s = 1765,6s). Die erstellten Zugriffsstrukturen belegten insgesamt ca. 770 MB Speicherplatz.

<table>
<thead>
<tr>
<th>Chr.</th>
<th>real (s)</th>
<th>user (s)</th>
<th>sys (s)</th>
<th>%cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>148,5</td>
<td>92,8</td>
<td>12,4</td>
<td>70,84</td>
</tr>
<tr>
<td>2</td>
<td>144,5</td>
<td>88,4</td>
<td>12,3</td>
<td>69,68</td>
</tr>
<tr>
<td>3</td>
<td>119,1</td>
<td>73,7</td>
<td>9,8</td>
<td>70,10</td>
</tr>
<tr>
<td>4</td>
<td>114,5</td>
<td>69,6</td>
<td>8,9</td>
<td>68,55</td>
</tr>
<tr>
<td>5</td>
<td>107,6</td>
<td>66,7</td>
<td>8,6</td>
<td>69,98</td>
</tr>
<tr>
<td>6</td>
<td>101,9</td>
<td>63,2</td>
<td>8,2</td>
<td>70,06</td>
</tr>
<tr>
<td>7</td>
<td>94,2</td>
<td>57,8</td>
<td>7,9</td>
<td>69,74</td>
</tr>
<tr>
<td>8</td>
<td>85,4</td>
<td>53,5</td>
<td>8,0</td>
<td>72,01</td>
</tr>
<tr>
<td>9</td>
<td>71,6</td>
<td>53,7</td>
<td>5,8</td>
<td>83,10</td>
</tr>
<tr>
<td>10</td>
<td>75,9</td>
<td>49,4</td>
<td>6,8</td>
<td>74,04</td>
</tr>
<tr>
<td>11</td>
<td>76,0</td>
<td>49,3</td>
<td>6,7</td>
<td>73,68</td>
</tr>
<tr>
<td>12</td>
<td>74,0</td>
<td>48,5</td>
<td>6,4</td>
<td>74,18</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Chr.</th>
<th>real (s)</th>
<th>user (s)</th>
<th>sys (s)</th>
<th>%cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td>13</td>
<td>65,4</td>
<td>42,9</td>
<td>5,3</td>
<td>73,70</td>
</tr>
<tr>
<td>14</td>
<td>61,4</td>
<td>40,7</td>
<td>5,4</td>
<td>75,08</td>
</tr>
<tr>
<td>15</td>
<td>58,3</td>
<td>38,5</td>
<td>4,9</td>
<td>74,44</td>
</tr>
<tr>
<td>16</td>
<td>51,5</td>
<td>33,9</td>
<td>4,4</td>
<td>74,36</td>
</tr>
<tr>
<td>17</td>
<td>46,6</td>
<td>30,7</td>
<td>4,1</td>
<td>74,67</td>
</tr>
<tr>
<td>18</td>
<td>44,3</td>
<td>27,8</td>
<td>4,0</td>
<td>71,78</td>
</tr>
<tr>
<td>19</td>
<td>37,7</td>
<td>24,3</td>
<td>3,3</td>
<td>73,20</td>
</tr>
<tr>
<td>20</td>
<td>37,4</td>
<td>23,2</td>
<td>3,5</td>
<td>71,39</td>
</tr>
<tr>
<td>21</td>
<td>29,9</td>
<td>19,2</td>
<td>2,5</td>
<td>72,57</td>
</tr>
<tr>
<td>22</td>
<td>24,2</td>
<td>19,5</td>
<td>1,4</td>
<td>86,36</td>
</tr>
<tr>
<td>X</td>
<td>70,4</td>
<td>56,7</td>
<td>4,0</td>
<td>86,22</td>
</tr>
<tr>
<td>Y</td>
<td>26,3</td>
<td>21,8</td>
<td>1,3</td>
<td>87,83</td>
</tr>
</tbody>
</table>

\[
\begin{array}{cccc}
\text{Chr.} & \text{real (s)} & \text{user (s)} & \text{sys (s)} & \%cpu \\
\hline
1 & 148,5 & 92,8 & 12,4 & 70,84 \\
2 & 144,5 & 88,4 & 12,3 & 69,68 \\
3 & 119,1 & 73,7 & 9,8 & 70,10 \\
4 & 114,5 & 69,6 & 8,9 & 68,55 \\
5 & 107,6 & 66,7 & 8,6 & 69,98 \\
6 & 101,9 & 63,2 & 8,2 & 70,06 \\
7 & 94,2 & 57,8 & 7,9 & 69,74 \\
8 & 85,4 & 53,5 & 8,0 & 72,01 \\
9 & 71,6 & 53,7 & 5,8 & 83,10 \\
10 & 75,9 & 49,4 & 6,8 & 74,04 \\
11 & 76,0 & 49,3 & 6,7 & 73,68 \\
12 & 74,0 & 48,5 & 6,4 & 74,18 \\
\hline
\text{13} & 65,4 & 42,9 & 5,3 & 73,70 \\
\text{14} & 61,4 & 40,7 & 5,4 & 75,08 \\
\text{15} & 58,3 & 38,5 & 4,9 & 74,44 \\
\text{16} & 51,5 & 33,9 & 4,4 & 74,36 \\
\text{17} & 46,6 & 30,7 & 4,1 & 74,67 \\
\text{18} & 44,3 & 27,8 & 4,0 & 71,78 \\
\text{19} & 37,7 & 24,3 & 3,3 & 73,20 \\
\text{20} & 37,4 & 23,2 & 3,5 & 71,39 \\
\text{21} & 29,9 & 19,2 & 2,5 & 72,57 \\
\text{22} & 24,2 & 19,5 & 1,4 & 86,36 \\
\text{X} & 70,4 & 56,7 & 4,0 & 86,22 \\
\text{Y} & 26,3 & 21,8 & 1,3 & 87,83 \\
\hline
\text{\textit{∑}} & \text{1213,2} & \text{766,6} & \text{101,8} & \text{-} \\
\text{\textit{∑}} & \text{553,4} & \text{379,2} & \text{44,1} & \text{-}
\end{array}
\]

Tabelle B.1: Laufzeiten und proz. CPU - Auslastung BLAST - Vorverarbeitung

B.2 VMATCH (ENSEMBL - VERSION 18_34)

Die Vorverarbeitung erfolgte chromosomenweise durch den Aufruf

```
mkvtree -db <inputfile> - dna -allout -pl -u
```

\[1\text{s. VMATCH - Handbuch}\]

ANHANG B. TABELLEN MIT LAUFZEITEN

Die Prozessorauslastung nach der Formel 

\[ \text{Hauptanteil der Zeit verursacht das Sortieren der Suffixe und der Buckets; nur ein geringer Teil entfällt auf das Schreiben in Dateien. Zum Hinterlegen der Zugriffsstrukturen wird pro Chromosom zusätzlich ca. der 52 - 56fache Speicherplatz verbraucht, insgesamt (ohne FASTA - DNA - Sequenzen) ca. 41998 MB.} \]

Im Vergleich zu BLAST braucht VMATCH bei der Vorverarbeitung in Bezug auf real bzw. user + sys pro Chromosom durchschnittlich die 8 - 10fache bzw. 8 - 11fache Zeit. Den Hauptanteil der Zeit verursacht das Sortieren der Suffixe und der Buckets; nur ein geringer Teil entfällt auf das Schreiben in Dateien.\n
Die Tabellen B.3 bis B.8 geben für die Tandems (B.3 auf Seite 125), supermaximalen Repeats (B.4 auf Seite 126), Repeats (B.5, B.6 auf Seite 126) und Palindrome (B.7, B.8 auf Seite 127) jeweils die Anzahl der Ergebnisse, die benötigte Laufzeit für die Berechnung und die Prozessorauslastung nach der Formel

<table>
<thead>
<tr>
<th>Chr.</th>
<th>real (s)</th>
<th>user (s)</th>
<th>sys (s)</th>
<th>%cpu</th>
<th>Chr.</th>
<th>real (s)</th>
<th>user (s)</th>
<th>sys (s)</th>
<th>%cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1406,7</td>
<td>1031,5</td>
<td>115,3</td>
<td>81,52</td>
<td>13</td>
<td>542,2</td>
<td>373,5</td>
<td>40,0</td>
<td>76,26</td>
</tr>
<tr>
<td>2</td>
<td>1617,5</td>
<td>1257,5</td>
<td>117,5</td>
<td>85,00</td>
<td>14</td>
<td>498,1</td>
<td>340,7</td>
<td>37,9</td>
<td>76,00</td>
</tr>
<tr>
<td>3</td>
<td>1117,2</td>
<td>818,7</td>
<td>82,1</td>
<td>80,63</td>
<td>15</td>
<td>501,7</td>
<td>350,1</td>
<td>37,7</td>
<td>77,29</td>
</tr>
<tr>
<td>4</td>
<td>1090,5</td>
<td>806,3</td>
<td>79,1</td>
<td>81,19</td>
<td>16</td>
<td>462,0</td>
<td>329,2</td>
<td>31,5</td>
<td>78,07</td>
</tr>
<tr>
<td>5</td>
<td>1044,8</td>
<td>774,1</td>
<td>66,7</td>
<td>80,47</td>
<td>17</td>
<td>406,4</td>
<td>287,6</td>
<td>26,6</td>
<td>77,31</td>
</tr>
<tr>
<td>6</td>
<td>960,3</td>
<td>703,4</td>
<td>63,0</td>
<td>79,80</td>
<td>18</td>
<td>355,3</td>
<td>242,6</td>
<td>25,3</td>
<td>75,40</td>
</tr>
<tr>
<td>7</td>
<td>883,3</td>
<td>646,2</td>
<td>55,2</td>
<td>79,40</td>
<td>19</td>
<td>300,9</td>
<td>206,0</td>
<td>22,6</td>
<td>75,97</td>
</tr>
<tr>
<td>8</td>
<td>811,8</td>
<td>592,0</td>
<td>49,8</td>
<td>79,05</td>
<td>20</td>
<td>287,4</td>
<td>194,2</td>
<td>21,2</td>
<td>74,94</td>
</tr>
<tr>
<td>9</td>
<td>873,7</td>
<td>662,7</td>
<td>50,3</td>
<td>81,60</td>
<td>21</td>
<td>197,3</td>
<td>125,1</td>
<td>17,4</td>
<td>72,22</td>
</tr>
<tr>
<td>10</td>
<td>789,8</td>
<td>585,6</td>
<td>46,4</td>
<td>80,02</td>
<td>22</td>
<td>214,5</td>
<td>137,9</td>
<td>18,5</td>
<td>72,91</td>
</tr>
<tr>
<td>11</td>
<td>741,8</td>
<td>539,3</td>
<td>45,6</td>
<td>78,84</td>
<td>X</td>
<td>915,3</td>
<td>684,8</td>
<td>51,7</td>
<td>80,46</td>
</tr>
<tr>
<td>12</td>
<td>690,5</td>
<td>500,9</td>
<td>43,0</td>
<td>78,76</td>
<td>Y</td>
<td>379,9</td>
<td>292,4</td>
<td>22,8</td>
<td>84,98</td>
</tr>
<tr>
<td>( \sum )</td>
<td>19207,9</td>
<td>18912,8</td>
<td>1840</td>
<td>-</td>
<td>( \sum )</td>
<td>5052,0</td>
<td>3564,1</td>
<td>353,2</td>
<td>-</td>
</tr>
</tbody>
</table>

Tabelle B.2: Laufzeiten und prozentuale CPU - Auslastung VMATCH - Vorverarbeitung

<table>
<thead>
<tr>
<th>Chr.</th>
<th>real (s)</th>
<th>user (s)</th>
<th>sys (s)</th>
<th>%cpu</th>
<th>Chr.</th>
<th>real (s)</th>
<th>user (s)</th>
<th>sys (s)</th>
<th>%cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3536,1</td>
<td>282,8</td>
<td>24,3</td>
<td>21,10</td>
<td>13</td>
<td>309</td>
<td>98,9</td>
<td>12,1</td>
<td>9,7</td>
</tr>
<tr>
<td>2</td>
<td>263,1</td>
<td>29,7</td>
<td>24,9</td>
<td>20,75</td>
<td>14</td>
<td>228</td>
<td>93,5</td>
<td>11,3</td>
<td>9,3</td>
</tr>
<tr>
<td>3</td>
<td>202,4</td>
<td>23,6</td>
<td>19,9</td>
<td>21,49</td>
<td>15</td>
<td>217</td>
<td>91,6</td>
<td>11,7</td>
<td>9,0</td>
</tr>
<tr>
<td>4</td>
<td>201,8</td>
<td>22,1</td>
<td>20,3</td>
<td>21,01</td>
<td>16</td>
<td>360</td>
<td>99,2</td>
<td>11,2</td>
<td>9,1</td>
</tr>
<tr>
<td>5</td>
<td>196,0</td>
<td>22,7</td>
<td>18,4</td>
<td>20,96</td>
<td>17</td>
<td>415</td>
<td>85,5</td>
<td>10,1</td>
<td>9,0</td>
</tr>
<tr>
<td>6</td>
<td>181,7</td>
<td>19,9</td>
<td>18,3</td>
<td>21,02</td>
<td>18</td>
<td>249</td>
<td>74,9</td>
<td>8,7</td>
<td>7,0</td>
</tr>
<tr>
<td>7</td>
<td>171,6</td>
<td>19,2</td>
<td>17,6</td>
<td>21,44</td>
<td>19</td>
<td>365</td>
<td>62,2</td>
<td>7,3</td>
<td>6,7</td>
</tr>
<tr>
<td>8</td>
<td>150,3</td>
<td>16,9</td>
<td>15,7</td>
<td>21,22</td>
<td>20</td>
<td>209</td>
<td>59,7</td>
<td>6,5</td>
<td>6,1</td>
</tr>
<tr>
<td>9</td>
<td>131,1</td>
<td>16,2</td>
<td>13,6</td>
<td>22,73</td>
<td>21</td>
<td>138</td>
<td>36,5</td>
<td>5,0</td>
<td>3,5</td>
</tr>
<tr>
<td>10</td>
<td>143,4</td>
<td>16,9</td>
<td>13,3</td>
<td>21,12</td>
<td>22</td>
<td>134</td>
<td>39,9</td>
<td>5,2</td>
<td>3,8</td>
</tr>
<tr>
<td>11</td>
<td>138,4</td>
<td>15,7</td>
<td>13,8</td>
<td>21,31</td>
<td>X</td>
<td>460</td>
<td>125,6</td>
<td>18,6</td>
<td>18,2</td>
</tr>
<tr>
<td>12</td>
<td>135,8</td>
<td>15,4</td>
<td>14,1</td>
<td>21,72</td>
<td>Y</td>
<td>42</td>
<td>27,9</td>
<td>5,3</td>
<td>2,9</td>
</tr>
<tr>
<td>( \sum )</td>
<td>2164,3</td>
<td>146,5</td>
<td>213,6</td>
<td>-</td>
<td>( \sum )</td>
<td>895,4</td>
<td>113,0</td>
<td>94,3</td>
<td>-</td>
</tr>
</tbody>
</table>

Tabelle B.3: Anz. Tandems, VMATCH-Laufzeiten (Sek.) und Prozessorauslastung
und Palindromdurchl"aufen l"asst sich mit der Berechnungsintensit"at erkl"aren, so auch die
- Verz"ogerungen. Die ansteigende Prozessorauslastung von Tandems zu supermaximal Re-
verse complement zur Palindrombestimmung erst zur Laufzeit berechnet wird.

Das Laufzeitverh"altnis zwischen Repeats und Palindromen deutet darauf hin, dass das re-
Minuten (Repeat) zu 434 Minuten (Palindrom).

Die Gesamtlaufzeit der Tandemberechnung ist mit ca. 50 Minuten (real) und der
supermaximal Repeatberechnung mit ca. 14 Minuten (real) in bezug auf die Genomgr"o-
fenere die hohe Prozessorauslastung bei den beiden Repeat-
und Palindromdurchl"aufen l"asst sich mit der Berechnungsintensit"at erkennen, so auch die
Laufzeitunterschiede (real) von 54 Minuten (Repeat) zu 365 Minuten (Palindrom) und 101
Minuten (Repeat) zu 434 Minuten (Palindrom).

Das Laufzeitverh"altnis zwischen Repeats und Palindromen deutet darauf hin, dass das re-
verse complement zur Palindrombestimmung erst zur Laufzeit berechnet wird.

### Tabelle B.4: Anz. supermax. Repeats, VMATCH-Laufzeiten (Sek.) und Prozessorauslastung

\[
\%\text{cpu} = \frac{100(\text{user}+\text{sys})}{\text{real}}
\]

wieder. Die Gesamtauflaufzeit der Tandemberechnung ist mit ca. 50 Minuten (real) und der
supermaximal Repeatberechnung mit ca. 14 Minuten (real) in bezug auf die Genomgr"o-
fenere die hohe Prozessorauslastung bei den beiden Repeat-
und Palindromdurchl"aufen l"asst sich mit der Berechnungsintensit"at erkennen, so auch die
Laufzeitunterschiede (real) von 54 Minuten (Repeat) zu 365 Minuten (Palindrom) und 101
Minuten (Repeat) zu 434 Minuten (Palindrom).

Das Laufzeitverh"altnis zwischen Repeats und Palindromen deutet darauf hin, dass das re-
verse complement zur Palindrombestimmung erst zur Laufzeit berechnet wird.

| Chr. | Anz. | real | user | sys | %cpu | Chr. | Anz. | real | user | sys | %cpu |
|------|------|------|------|-----|-----|------|------|------|------|-----|-----|-----|
| 1    | 17877| 67.3 | 16.2 | 9.0 | 37.44| 13   | 5007 | 28.5 | 7.1  | 3.5 | 37.19|
| 2    | 18617| 69.0 | 15.8 | 8.7 | 35.50| 14   | 5261 | 26.5 | 6.3  | 3.7 | 37.73|
| 3    | 14179| 55.2 | 12.8 | 7.3 | 36.41| 15   | 7572 | 26.7 | 6.8  | 3.8 | 39.70|
| 4    | 15557| 55.5 | 12.3 | 7.0 | 34.77| 16   | 9534 | 26.4 | 6.2  | 4.0 | 38.63|
| 5    | 15628| 52.2 | 11.7 | 7.2 | 36.60| 17   | 5871 | 23.2 | 5.2  | 3.2 | 36.20|
| 6    | 12480| 48.2 | 10.6 | 6.6 | 35.68| 18   | 3835 | 20.8 | 4.7  | 2.8 | 36.06|
| 7    | 13212| 45.9 | 11.1 | 6.6 | 38.56| 19   | 3792 | 17.0 | 3.9  | 2.1 | 35.29|
| 8    | 10366| 41.1 | 9.2  | 5.4 | 35.52| 20   | 2417 | 16.8 | 3.6  | 2.0 | 33.33|
| 9    | 12865| 37.8 | 9.7  | 5.5 | 40.21| 21   | 1343 | 11.4 | 2.9  | 1.5 | 38.59|
| 10   | 11015| 40.3 | 9.4  | 5.6 | 37.22| 22   | 2763 | 11.8 | 2.9  | 1.6 | 38.13|
| 11   | 9607  | 37.5 | 9.0  | 4.9 | 37.06|  X   | 17112| 48.7 | 9.7  | 7.0 | 34.29|
| 12   | 8231  | 36.6 | 7.9  | 4.5 | 33.87|  Y   | 3377 | 10.6 | 3.0  | 1.5 | 42.45|
| ∑    | 538.4 | 125.1| 71.7 | -   | -   |      | 268.4| 62.3 | 40.9 | -  | -   |

Tabelle B.5: Anz. Repeats, VMATCH-Laufzeiten (Sek.) und Prozessorauslastung (I)
### Tabelle B.6: Anz. Repeats, VMATCH-Laufzeiten (Sek.) und Prozessorauslastung (II)

<table>
<thead>
<tr>
<th>Chr.</th>
<th>Anz.</th>
<th>real</th>
<th>user</th>
<th>sys</th>
<th>%cpu</th>
<th>Chr.</th>
<th>Anz.</th>
<th>real</th>
<th>user</th>
<th>sys</th>
<th>%cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3737010</td>
<td>575.9</td>
<td>476.9</td>
<td>98.9</td>
<td>87.9</td>
<td>13</td>
<td>638827</td>
<td>137.9</td>
<td>92.5</td>
<td>13.0</td>
<td>76.5</td>
</tr>
<tr>
<td>2</td>
<td>4754577</td>
<td>620.5</td>
<td>519.2</td>
<td>39.9</td>
<td>88.6</td>
<td>14</td>
<td>691830</td>
<td>137.9</td>
<td>92.5</td>
<td>13.0</td>
<td>78.7</td>
</tr>
<tr>
<td>3</td>
<td>4433211</td>
<td>532.4</td>
<td>446.0</td>
<td>37.2</td>
<td>88.8</td>
<td>15</td>
<td>430460</td>
<td>120.5</td>
<td>81.4</td>
<td>10.8</td>
<td>78.5</td>
</tr>
<tr>
<td>4</td>
<td>4118854</td>
<td>526.3</td>
<td>476.8</td>
<td>89.2</td>
<td>29.8</td>
<td>16</td>
<td>468581</td>
<td>108.4</td>
<td>73.8</td>
<td>10.5</td>
<td>77.7</td>
</tr>
<tr>
<td>5</td>
<td>4229095</td>
<td>492.1</td>
<td>413.2</td>
<td>42.9</td>
<td>88.8</td>
<td>17</td>
<td>159301</td>
<td>113.7</td>
<td>81.9</td>
<td>8.9</td>
<td>79.8</td>
</tr>
<tr>
<td>6</td>
<td>3036792</td>
<td>396.4</td>
<td>322.9</td>
<td>37.2</td>
<td>87.0</td>
<td>18</td>
<td>419834</td>
<td>93.8</td>
<td>63.3</td>
<td>8.7</td>
<td>76.7</td>
</tr>
<tr>
<td>7</td>
<td>2016717</td>
<td>322.8</td>
<td>256.8</td>
<td>71.5</td>
<td>85.9</td>
<td>19</td>
<td>103873</td>
<td>89.5</td>
<td>64.7</td>
<td>6.8</td>
<td>79.8</td>
</tr>
<tr>
<td>8</td>
<td>2283522</td>
<td>303.8</td>
<td>242.8</td>
<td>91.4</td>
<td>85.9</td>
<td>20</td>
<td>143752</td>
<td>65.3</td>
<td>41.3</td>
<td>7.0</td>
<td>73.9</td>
</tr>
<tr>
<td>9</td>
<td>1226994</td>
<td>217.5</td>
<td>162.2</td>
<td>37.2</td>
<td>81.8</td>
<td>21</td>
<td>39294</td>
<td>35.9</td>
<td>24.1</td>
<td>4.7</td>
<td>67.4</td>
</tr>
<tr>
<td>10</td>
<td>1330067</td>
<td>232.7</td>
<td>171.6</td>
<td>24.9</td>
<td>83.7</td>
<td>22</td>
<td>24227</td>
<td>42.4</td>
<td>24.1</td>
<td>4.7</td>
<td>67.9</td>
</tr>
</tbody>
</table>

| ∑    | 4395.2 | 3595.2 | 243.2 | -   |      | ∑    | 1672.0 | 1238.2 | 119.9 | -   |

B.3 Datenbank / C - Programm(e)

### Tabelle B.7: Anz. Palindrome, VMATCH-Laufzeiten (Sek.) und Prozessorauslastung (I)

<table>
<thead>
<tr>
<th>Chr.</th>
<th>Anz.</th>
<th>real</th>
<th>user</th>
<th>sys</th>
<th>%cpu</th>
<th>Chr.</th>
<th>Anz.</th>
<th>real</th>
<th>user</th>
<th>sys</th>
<th>%cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1399573</td>
<td>2182.4</td>
<td>2007.5</td>
<td>36.5</td>
<td>93.6</td>
<td>13</td>
<td>248338</td>
<td>569.3</td>
<td>508.2</td>
<td>12.6</td>
<td>91.4</td>
</tr>
<tr>
<td>2</td>
<td>1792342</td>
<td>2240.2</td>
<td>2059.0</td>
<td>31.2</td>
<td>93.5</td>
<td>14</td>
<td>39294</td>
<td>35.9</td>
<td>19.0</td>
<td>5.2</td>
<td>67.8</td>
</tr>
<tr>
<td>3</td>
<td>1618999</td>
<td>1872.0</td>
<td>1716.2</td>
<td>27.5</td>
<td>93.3</td>
<td>15</td>
<td>170644</td>
<td>569.3</td>
<td>508.2</td>
<td>12.6</td>
<td>91.4</td>
</tr>
<tr>
<td>4</td>
<td>1829664</td>
<td>1687.3</td>
<td>1534.9</td>
<td>31.2</td>
<td>92.8</td>
<td>16</td>
<td>113738</td>
<td>576.4</td>
<td>519.5</td>
<td>12.4</td>
<td>92.3</td>
</tr>
<tr>
<td>5</td>
<td>1510723</td>
<td>1426.3</td>
<td>1289.7</td>
<td>28.3</td>
<td>92.4</td>
<td>17</td>
<td>64149</td>
<td>512.9</td>
<td>462.6</td>
<td>10.9</td>
<td>92.3</td>
</tr>
<tr>
<td>6</td>
<td>1067843</td>
<td>1295.9</td>
<td>1169.1</td>
<td>22.9</td>
<td>91.9</td>
<td>18</td>
<td>161861</td>
<td>457.3</td>
<td>405.8</td>
<td>11.3</td>
<td>91.2</td>
</tr>
<tr>
<td>7</td>
<td>759466</td>
<td>1153.4</td>
<td>1032.5</td>
<td>23.4</td>
<td>91.5</td>
<td>19</td>
<td>41480</td>
<td>326.4</td>
<td>284.3</td>
<td>8.3</td>
<td>86.9</td>
</tr>
<tr>
<td>8</td>
<td>827741</td>
<td>924.9</td>
<td>825.4</td>
<td>20.1</td>
<td>89.9</td>
<td>20</td>
<td>56761</td>
<td>322.2</td>
<td>279.2</td>
<td>9.4</td>
<td>89.5</td>
</tr>
<tr>
<td>9</td>
<td>457606</td>
<td>892.4</td>
<td>782.5</td>
<td>20.1</td>
<td>89.9</td>
<td>21</td>
<td>15276</td>
<td>164.7</td>
<td>133.8</td>
<td>6.7</td>
<td>85.3</td>
</tr>
<tr>
<td>10</td>
<td>478012</td>
<td>1030.5</td>
<td>918.7</td>
<td>21.5</td>
<td>91.2</td>
<td>22</td>
<td>12427</td>
<td>180.8</td>
<td>148.7</td>
<td>6.6</td>
<td>85.6</td>
</tr>
<tr>
<td>11</td>
<td>732450</td>
<td>1013.0</td>
<td>903.3</td>
<td>22.6</td>
<td>91.4</td>
<td>23</td>
<td>2259751</td>
<td>1454.5</td>
<td>1241.6</td>
<td>30.0</td>
<td>87.4</td>
</tr>
<tr>
<td>12</td>
<td>568027</td>
<td>1021.1</td>
<td>938.8</td>
<td>18.9</td>
<td>93.7</td>
<td>24</td>
<td>160273</td>
<td>346.4</td>
<td>28.0</td>
<td>5.4</td>
<td>71.9</td>
</tr>
</tbody>
</table>

| ∑    | 15998.9 | 14528.3 | 295.9 | -   |      | ∑    | 5909.8  | 5165.3 | 119.9 | -   |
ANHANG B. TABELLEN MIT LAUFZEITEN

<table>
<thead>
<tr>
<th>Chr.</th>
<th>Anz.</th>
<th>real (s)</th>
<th>user (s)</th>
<th>sys (s)</th>
<th>%usr</th>
<th>%sys</th>
<th>%wio</th>
<th>%idle</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3727198</td>
<td>2663.7</td>
<td>2571.2</td>
<td>33.9</td>
<td>97.8</td>
<td>12.7</td>
<td>96.0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>4737182</td>
<td>2845.8</td>
<td>2741.1</td>
<td>36.1</td>
<td>97.5</td>
<td>11.6</td>
<td>96.3</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>4445015</td>
<td>2252.5</td>
<td>2169.0</td>
<td>29.6</td>
<td>97.6</td>
<td>25.0</td>
<td>96.5</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>5122143</td>
<td>2196.8</td>
<td>2115.6</td>
<td>30.0</td>
<td>96.5</td>
<td>12.7</td>
<td>96.6</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4235226</td>
<td>1934.3</td>
<td>1824.3</td>
<td>36.1</td>
<td>97.5</td>
<td>11.6</td>
<td>96.1</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>3005440</td>
<td>1743.3</td>
<td>1670.7</td>
<td>25.0</td>
<td>97.2</td>
<td>25.0</td>
<td>96.1</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>1992035</td>
<td>1538.9</td>
<td>1472.2</td>
<td>22.9</td>
<td>97.1</td>
<td>22.9</td>
<td>96.1</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>2278256</td>
<td>1353.7</td>
<td>1290.9</td>
<td>19.7</td>
<td>96.8</td>
<td>19.7</td>
<td>95.2</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>1305471</td>
<td>1226.2</td>
<td>1169.7</td>
<td>19.9</td>
<td>97.0</td>
<td>19.9</td>
<td>97.6</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1555358</td>
<td>1180.3</td>
<td>1124.3</td>
<td>18.3</td>
<td>97.4</td>
<td>18.3</td>
<td>97.6</td>
<td></td>
</tr>
<tr>
<td>∑</td>
<td></td>
<td>19515.0</td>
<td>18726.7</td>
<td>251.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Tabelle B.8: Anz. Palindrome, VMATCH-Laufzeiten (Sek.) und Prozessorauslastung (II)

<table>
<thead>
<tr>
<th>Chr.</th>
<th>Anz.</th>
<th>real (s)</th>
<th>user (s)</th>
<th>sys (s)</th>
<th>%usr</th>
<th>%sys</th>
<th>%wio</th>
<th>%idle</th>
</tr>
</thead>
<tbody>
<tr>
<td>hs_masked_rep</td>
<td>204.69</td>
<td>0.32</td>
<td>1.18</td>
<td>1</td>
<td>1</td>
<td>21</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>hs_masked_pal</td>
<td>190.97</td>
<td>0.24</td>
<td>1.18</td>
<td>1</td>
<td>1</td>
<td>20</td>
<td>77</td>
<td></td>
</tr>
<tr>
<td>pt_unmasked_rep</td>
<td>12.048,51</td>
<td>0.34</td>
<td>1.28</td>
<td>1</td>
<td>0</td>
<td>24</td>
<td>74</td>
<td></td>
</tr>
<tr>
<td>pt_unmasked_pal</td>
<td>6.174,09</td>
<td>0.44</td>
<td>1.36</td>
<td>2</td>
<td>2</td>
<td>34</td>
<td>62</td>
<td></td>
</tr>
<tr>
<td>hs_unmasked_rep</td>
<td>27.624,79</td>
<td>0.49</td>
<td>1.86</td>
<td>2</td>
<td>2</td>
<td>35</td>
<td>61</td>
<td></td>
</tr>
<tr>
<td>hs_unmasked_pal</td>
<td>32.371,76</td>
<td>0.49</td>
<td>1.56</td>
<td>1</td>
<td>2</td>
<td>26</td>
<td>71</td>
<td></td>
</tr>
</tbody>
</table>

Tabelle B.9: Laufzeiten zur Redundanzerkennung mit SQL PL

<table>
<thead>
<tr>
<th>Chr.</th>
<th>Anz.</th>
<th>real (s)</th>
<th>user (s)</th>
<th>sys (s)</th>
<th>%usr</th>
<th>%sys</th>
<th>%wio</th>
<th>%idle</th>
</tr>
</thead>
<tbody>
<tr>
<td>hs_masked_rep</td>
<td>7.30</td>
<td>0.49</td>
<td>0.56</td>
<td>10</td>
<td>1</td>
<td>2</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>hs_masked_pal</td>
<td>7.60</td>
<td>0.53</td>
<td>0.59</td>
<td>12</td>
<td>1</td>
<td>1</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>pt_unmasked_rep</td>
<td>598.50</td>
<td>22.11</td>
<td>572.89</td>
<td>12</td>
<td>1</td>
<td>0</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>pt_unmasked_pal</td>
<td>579.46</td>
<td>21.24</td>
<td>554.60</td>
<td>12</td>
<td>1</td>
<td>0</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>hs_unmasked_rep</td>
<td>1.961,41</td>
<td>67.19</td>
<td>1886.31</td>
<td>12</td>
<td>1</td>
<td>0</td>
<td>87</td>
<td></td>
</tr>
<tr>
<td>hs_unmasked_pal</td>
<td>1.862,06</td>
<td>56.75</td>
<td>1798.75</td>
<td>12</td>
<td>1</td>
<td>0</td>
<td>87</td>
<td></td>
</tr>
</tbody>
</table>

Tabelle B.10: Laufzeiten zur Redundanzerkennung mit C - Programm
Anhang C

Tabellen mit Anzahl der Duplikate

C.1 Tabellen zu den unbestimmten DNA - Sequenzen

C.1.1 *homo sapiens* (unmasked)

<table>
<thead>
<tr>
<th>Chr.</th>
<th>DNA - Gesamtlänge</th>
<th>kumul. Gapsize</th>
<th>Anz. Gaps</th>
<th>Gaplänge in %</th>
<th>Chr.</th>
<th>DNA - Gesamtlänge</th>
<th>kumul. Gapsize</th>
<th>Anz. Gaps</th>
<th>Gaplänge in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>246127941</td>
<td>24565000</td>
<td>65</td>
<td>9.98</td>
<td>13</td>
<td>113042980</td>
<td>17483000</td>
<td>6</td>
<td>15.46</td>
</tr>
<tr>
<td>2</td>
<td>243615958</td>
<td>6074355</td>
<td>30</td>
<td>2.49</td>
<td>14</td>
<td>105311216</td>
<td>18120000</td>
<td>2</td>
<td>17.20</td>
</tr>
<tr>
<td>3</td>
<td>199344050</td>
<td>35002</td>
<td>3</td>
<td>0.02</td>
<td>15</td>
<td>100256656</td>
<td>18997000</td>
<td>11</td>
<td>18.94</td>
</tr>
<tr>
<td>4</td>
<td>191731959</td>
<td>4890000</td>
<td>17</td>
<td>2.55</td>
<td>16</td>
<td>90041932</td>
<td>10109503</td>
<td>15</td>
<td>11.22</td>
</tr>
<tr>
<td>5</td>
<td>181034922</td>
<td>3482100</td>
<td>13</td>
<td>1.92</td>
<td>17</td>
<td>81860266</td>
<td>4182522</td>
<td>13</td>
<td>5.10</td>
</tr>
<tr>
<td>6</td>
<td>170914576</td>
<td>3658001</td>
<td>11</td>
<td>2.14</td>
<td>18</td>
<td>76115139</td>
<td>1461098</td>
<td>5</td>
<td>1.91</td>
</tr>
<tr>
<td>7</td>
<td>158545518</td>
<td>3869000</td>
<td>12</td>
<td>2.44</td>
<td>19</td>
<td>63811651</td>
<td>8026000</td>
<td>5</td>
<td>12.57</td>
</tr>
<tr>
<td>8</td>
<td>146308819</td>
<td>3960900</td>
<td>13</td>
<td>2.70</td>
<td>20</td>
<td>63741868</td>
<td>4316878</td>
<td>8</td>
<td>6.77</td>
</tr>
<tr>
<td>9</td>
<td>136372045</td>
<td>20748003</td>
<td>49</td>
<td>15.21</td>
<td>21</td>
<td>46976097</td>
<td>13051790</td>
<td>10</td>
<td>27.78</td>
</tr>
<tr>
<td>10</td>
<td>135037215</td>
<td>3864009</td>
<td>31</td>
<td>2.86</td>
<td>22</td>
<td>49396572</td>
<td>15044921</td>
<td>30</td>
<td>30.45</td>
</tr>
<tr>
<td>11</td>
<td>134482954</td>
<td>3574100</td>
<td>10</td>
<td>2.65</td>
<td>X</td>
<td>153692391</td>
<td>4477000</td>
<td>22</td>
<td>2.91</td>
</tr>
<tr>
<td>12</td>
<td>132078379</td>
<td>2252102</td>
<td>13</td>
<td>1.70</td>
<td>Y</td>
<td>50286555</td>
<td>27824921</td>
<td>7</td>
<td>55.33</td>
</tr>
</tbody>
</table>

Tabelle C.1: Ergebnisse zu unbestimmten Sequenzabschnitten
### C.1.2 pan troglodytes (unmasked)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>229575298</td>
<td>33643831</td>
<td>22648</td>
<td>14.65</td>
<td>14</td>
<td>97804244</td>
<td>26454691</td>
<td>7333</td>
<td>27.04</td>
</tr>
<tr>
<td>2</td>
<td>203813066</td>
<td>26961093</td>
<td>18074</td>
<td>13.22</td>
<td>15</td>
<td>106954593</td>
<td>28580375</td>
<td>8596</td>
<td>26.72</td>
</tr>
<tr>
<td>3</td>
<td>209662276</td>
<td>39639475</td>
<td>17896</td>
<td>18.90</td>
<td>16</td>
<td>101535987</td>
<td>34928614</td>
<td>7158</td>
<td>34.40</td>
</tr>
<tr>
<td>4</td>
<td>188378868</td>
<td>28558501</td>
<td>16422</td>
<td>15.16</td>
<td>17</td>
<td>73346066</td>
<td>8905790</td>
<td>6381</td>
<td>12.14</td>
</tr>
<tr>
<td>5</td>
<td>175429504</td>
<td>22643437</td>
<td>15440</td>
<td>12.91</td>
<td>18</td>
<td>83875239</td>
<td>22008764</td>
<td>8290</td>
<td>26.23</td>
</tr>
<tr>
<td>6</td>
<td>161576975</td>
<td>29011506</td>
<td>15207</td>
<td>17.95</td>
<td>19</td>
<td>61571712</td>
<td>15724541</td>
<td>8493</td>
<td>25.53</td>
</tr>
<tr>
<td>7</td>
<td>149542033</td>
<td>20089925</td>
<td>13511</td>
<td>13.43</td>
<td>20</td>
<td>65473740</td>
<td>12121913</td>
<td>6844</td>
<td>43.48</td>
</tr>
<tr>
<td>8</td>
<td>138322177</td>
<td>22356197</td>
<td>12541</td>
<td>16.16</td>
<td>21</td>
<td>76473740</td>
<td>12121913</td>
<td>6844</td>
<td>43.48</td>
</tr>
<tr>
<td>9</td>
<td>136640551</td>
<td>20707661</td>
<td>13139</td>
<td>15.15</td>
<td>22</td>
<td>47338174</td>
<td>17031948</td>
<td>3388</td>
<td>35.97</td>
</tr>
<tr>
<td>10</td>
<td>135301796</td>
<td>20089925</td>
<td>13511</td>
<td>13.43</td>
<td>13</td>
<td>113042980</td>
<td>60784808</td>
<td>150089</td>
<td>53.77</td>
</tr>
<tr>
<td>11</td>
<td>123086034</td>
<td>27299259</td>
<td>10931</td>
<td>22.17</td>
<td>14</td>
<td>105311216</td>
<td>59427684</td>
<td>138162</td>
<td>56.43</td>
</tr>
<tr>
<td>12</td>
<td>117159028</td>
<td>21825247</td>
<td>10207</td>
<td>18.62</td>
<td>15</td>
<td>100256656</td>
<td>56733285</td>
<td>129729</td>
<td>56.58</td>
</tr>
<tr>
<td>13</td>
<td>134309081</td>
<td>17750372</td>
<td>12063</td>
<td>13.21</td>
<td>16</td>
<td>90041932</td>
<td>49149761</td>
<td>141403</td>
<td>54.58</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tabelle C.2: Ergebnisse zu unbestimmten Sequenzabschnitten</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### C.1.3 homo sapiens (masked)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>246127941</td>
<td>362388</td>
<td>53.27</td>
<td>13</td>
<td>13</td>
<td>113042980</td>
<td>60784808</td>
<td>150089</td>
<td>53.77</td>
</tr>
<tr>
<td>2</td>
<td>243615958</td>
<td>369790</td>
<td>47.03</td>
<td>14</td>
<td>14</td>
<td>105311216</td>
<td>59427684</td>
<td>138162</td>
<td>56.43</td>
</tr>
<tr>
<td>3</td>
<td>199344050</td>
<td>309318</td>
<td>48.65</td>
<td>15</td>
<td>15</td>
<td>100256656</td>
<td>56733285</td>
<td>129729</td>
<td>56.58</td>
</tr>
<tr>
<td>4</td>
<td>191731959</td>
<td>293583</td>
<td>49.49</td>
<td>16</td>
<td>16</td>
<td>90041932</td>
<td>49149761</td>
<td>141403</td>
<td>54.58</td>
</tr>
<tr>
<td>5</td>
<td>181034922</td>
<td>276364</td>
<td>48.30</td>
<td>17</td>
<td>17</td>
<td>81860266</td>
<td>40904277</td>
<td>132860</td>
<td>49.96</td>
</tr>
<tr>
<td>6</td>
<td>170914576</td>
<td>259435</td>
<td>47.66</td>
<td>18</td>
<td>18</td>
<td>76115139</td>
<td>34660993</td>
<td>115482</td>
<td>45.53</td>
</tr>
<tr>
<td>7</td>
<td>158545518</td>
<td>243095</td>
<td>48.88</td>
<td>19</td>
<td>19</td>
<td>63811651</td>
<td>40201973</td>
<td>98733</td>
<td>63.00</td>
</tr>
<tr>
<td>8</td>
<td>146308819</td>
<td>223929</td>
<td>49.06</td>
<td>20</td>
<td>20</td>
<td>63741868</td>
<td>32937543</td>
<td>102842</td>
<td>51.67</td>
</tr>
<tr>
<td>9</td>
<td>136372045</td>
<td>187185</td>
<td>55.61</td>
<td>21</td>
<td>21</td>
<td>46976097</td>
<td>28467904</td>
<td>53262</td>
<td>60.60</td>
</tr>
<tr>
<td>10</td>
<td>135037215</td>
<td>207643</td>
<td>47.89</td>
<td>22</td>
<td>22</td>
<td>49396972</td>
<td>31424576</td>
<td>58284</td>
<td>63.61</td>
</tr>
<tr>
<td>11</td>
<td>134482954</td>
<td>206847</td>
<td>49.63</td>
<td>23</td>
<td>23</td>
<td>51092391</td>
<td>91576889</td>
<td>209163</td>
<td>59.58</td>
</tr>
<tr>
<td>12</td>
<td>132078379</td>
<td>214784</td>
<td>49.99</td>
<td>24</td>
<td>24</td>
<td>50286555</td>
<td>41659791</td>
<td>29648</td>
<td>82.84</td>
</tr>
</tbody>
</table>

Tabelle C.3: Ergebnisse zu unbestimmten Sequenzabschnitten
C.2 Anzahl aller Duplikate

C.2.1 *homo sapiens* (unmaskiert)

<table>
<thead>
<tr>
<th>Chr.</th>
<th>PID 1</th>
<th>PID 2</th>
<th>PID 3</th>
<th>PID 4</th>
<th>PID 5</th>
<th>PID 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1406647</td>
<td>3737010</td>
<td>1399573</td>
<td>3727198</td>
<td>536</td>
<td>17877</td>
</tr>
<tr>
<td>2</td>
<td>1794139</td>
<td>1829664</td>
<td>1792342</td>
<td>1794139</td>
<td>4754577</td>
<td>4737182</td>
</tr>
<tr>
<td>3</td>
<td>1622468</td>
<td>4433211</td>
<td>1618999</td>
<td>1510723</td>
<td>4229095</td>
<td>4445015</td>
</tr>
<tr>
<td>4</td>
<td>1836311</td>
<td>5118854</td>
<td>1829664</td>
<td>1510723</td>
<td>4235226</td>
<td>4235226</td>
</tr>
<tr>
<td>5</td>
<td>1512938</td>
<td>4229095</td>
<td>1510723</td>
<td>4235226</td>
<td>4737182</td>
<td>4445015</td>
</tr>
<tr>
<td>6</td>
<td>1076594</td>
<td>3036792</td>
<td>1067843</td>
<td>2283522</td>
<td>3005440</td>
<td>3005440</td>
</tr>
<tr>
<td>7</td>
<td>776815</td>
<td>2016717</td>
<td>759466</td>
<td>2283522</td>
<td>3005440</td>
<td>3005440</td>
</tr>
<tr>
<td>8</td>
<td>830854</td>
<td>2283522</td>
<td>827741</td>
<td>2283522</td>
<td>1992035</td>
<td>1992035</td>
</tr>
<tr>
<td>9</td>
<td>459245</td>
<td>1226994</td>
<td>462151</td>
<td>1231470</td>
<td>1231470</td>
<td>1231470</td>
</tr>
<tr>
<td>10</td>
<td>494206</td>
<td>1330067</td>
<td>478012</td>
<td>1305471</td>
<td>1305471</td>
<td>1305471</td>
</tr>
<tr>
<td>11</td>
<td>731014</td>
<td>2059308</td>
<td>732450</td>
<td>2061271</td>
<td>2061271</td>
<td>2061271</td>
</tr>
<tr>
<td>12</td>
<td>574496</td>
<td>1561314</td>
<td>568027</td>
<td>1555358</td>
<td>1555358</td>
<td>1555358</td>
</tr>
<tr>
<td>13</td>
<td>252882</td>
<td>638827</td>
<td>248338</td>
<td>626926</td>
<td>626926</td>
<td>626926</td>
</tr>
<tr>
<td>14</td>
<td>260585</td>
<td>691830</td>
<td>252982</td>
<td>682899</td>
<td>682899</td>
<td>682899</td>
</tr>
<tr>
<td>15</td>
<td>172634</td>
<td>430460</td>
<td>170644</td>
<td>427548</td>
<td>427548</td>
<td>427548</td>
</tr>
<tr>
<td>16</td>
<td>119096</td>
<td>258851</td>
<td>113738</td>
<td>251123</td>
<td>251123</td>
<td>251123</td>
</tr>
<tr>
<td>17</td>
<td>70715</td>
<td>159301</td>
<td>64149</td>
<td>148676</td>
<td>148676</td>
<td>148676</td>
</tr>
<tr>
<td>18</td>
<td>163496</td>
<td>419834</td>
<td>161861</td>
<td>415719</td>
<td>415719</td>
<td>415719</td>
</tr>
<tr>
<td>19</td>
<td>46717</td>
<td>103873</td>
<td>41480</td>
<td>96485</td>
<td>96485</td>
<td>96485</td>
</tr>
<tr>
<td>20</td>
<td>58526</td>
<td>143752</td>
<td>56761</td>
<td>141188</td>
<td>141188</td>
<td>141188</td>
</tr>
<tr>
<td>21</td>
<td>17052</td>
<td>39294</td>
<td>15276</td>
<td>36380</td>
<td>36380</td>
<td>36380</td>
</tr>
<tr>
<td>22</td>
<td>13226</td>
<td>24227</td>
<td>12427</td>
<td>23730</td>
<td>23730</td>
<td>23730</td>
</tr>
<tr>
<td>X</td>
<td>2272275</td>
<td>6604576</td>
<td>2259751</td>
<td>6587646</td>
<td>6587646</td>
<td>6587646</td>
</tr>
<tr>
<td>Y</td>
<td>56778</td>
<td>45131</td>
<td>160273</td>
<td>136522</td>
<td>136522</td>
<td>136522</td>
</tr>
</tbody>
</table>

Tabelle C.4: Anzahl Duplikate nach Klassen aufgeschlüsselt
C.2.2  *mus musculus* (unmaskiert)

<table>
<thead>
<tr>
<th>Chr.</th>
<th>PID 1</th>
<th>PID 3</th>
<th>PID 2</th>
<th>PID 4</th>
<th>PID 5</th>
<th>PID 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6680571</td>
<td>6676411</td>
<td>13643309</td>
<td>13638862</td>
<td>860</td>
<td>30029</td>
</tr>
<tr>
<td>2</td>
<td>4830703</td>
<td>4799432</td>
<td>9982654</td>
<td>9942248</td>
<td>1241</td>
<td>25369</td>
</tr>
<tr>
<td>3</td>
<td>5714849</td>
<td>5684483</td>
<td>11888871</td>
<td>11857252</td>
<td>840</td>
<td>25330</td>
</tr>
<tr>
<td>4</td>
<td>4675203</td>
<td>4644261</td>
<td>9746841</td>
<td>9727391</td>
<td>916</td>
<td>26257</td>
</tr>
<tr>
<td>5</td>
<td>300525</td>
<td>2966298</td>
<td>5943680</td>
<td>5894890</td>
<td>788</td>
<td>20152</td>
</tr>
<tr>
<td>6</td>
<td>3836950</td>
<td>3821174</td>
<td>7871399</td>
<td>7851127</td>
<td>830</td>
<td>20836</td>
</tr>
<tr>
<td>7</td>
<td>2469322</td>
<td>2455814</td>
<td>5155647</td>
<td>5139738</td>
<td>776</td>
<td>24075</td>
</tr>
<tr>
<td>8</td>
<td>2124566</td>
<td>2121598</td>
<td>4223710</td>
<td>4224447</td>
<td>746</td>
<td>15912</td>
</tr>
<tr>
<td>9</td>
<td>1393486</td>
<td>1382185</td>
<td>2765019</td>
<td>2756399</td>
<td>865</td>
<td>14042</td>
</tr>
<tr>
<td>10</td>
<td>551254</td>
<td>2548816</td>
<td>5127622</td>
<td>5122816</td>
<td>558</td>
<td>17087</td>
</tr>
<tr>
<td>11</td>
<td>986379</td>
<td>974400</td>
<td>1863079</td>
<td>1858183</td>
<td>848</td>
<td>12442</td>
</tr>
<tr>
<td>12</td>
<td>1868418</td>
<td>1856610</td>
<td>3839832</td>
<td>3830081</td>
<td>687</td>
<td>14582</td>
</tr>
<tr>
<td>13</td>
<td>1682418</td>
<td>1666729</td>
<td>3460519</td>
<td>3438174</td>
<td>548</td>
<td>15912</td>
</tr>
<tr>
<td>14</td>
<td>1907991</td>
<td>1901216</td>
<td>3897562</td>
<td>3890822</td>
<td>702</td>
<td>15558</td>
</tr>
<tr>
<td>15</td>
<td>1587757</td>
<td>1572491</td>
<td>3271712</td>
<td>3260796</td>
<td>715</td>
<td>12586</td>
</tr>
<tr>
<td>16</td>
<td>1482138</td>
<td>1473861</td>
<td>2967652</td>
<td>2957958</td>
<td>470</td>
<td>12361</td>
</tr>
<tr>
<td>17</td>
<td>887035</td>
<td>877265</td>
<td>1777864</td>
<td>1770167</td>
<td>576</td>
<td>11668</td>
</tr>
<tr>
<td>18</td>
<td>1181820</td>
<td>1181785</td>
<td>2427249</td>
<td>2424801</td>
<td>511</td>
<td>10245</td>
</tr>
<tr>
<td>19</td>
<td>251348</td>
<td>238014</td>
<td>494890</td>
<td>476567</td>
<td>359</td>
<td>5926</td>
</tr>
<tr>
<td>X</td>
<td>12604090</td>
<td>12541987</td>
<td>≥ 27500000</td>
<td>≥ 27500000</td>
<td>524</td>
<td>45126</td>
</tr>
<tr>
<td>Y</td>
<td>566891</td>
<td>571743</td>
<td>885256</td>
<td>888735</td>
<td>137</td>
<td>20998</td>
</tr>
</tbody>
</table>

Tabelle C.5: Anzahl Duplikate nach Klassen aufgeschlüsselt
### C.2.3 *pan troglodytes* (unmaskiert)

<table>
<thead>
<tr>
<th>Chr.</th>
<th>PID 1</th>
<th>PID 3</th>
<th>PID 2</th>
<th>PID 4</th>
<th>PID 5</th>
<th>PID 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>500336</td>
<td>496226</td>
<td>1299953</td>
<td>1301561</td>
<td>281</td>
<td>10155</td>
</tr>
<tr>
<td>2</td>
<td>676326</td>
<td>672561</td>
<td>1789258</td>
<td>1788117</td>
<td>276</td>
<td>10462</td>
</tr>
<tr>
<td>3</td>
<td>770984</td>
<td>767537</td>
<td>2065652</td>
<td>2067009</td>
<td>273</td>
<td>10477</td>
</tr>
<tr>
<td>4</td>
<td>631651</td>
<td>627160</td>
<td>1653233</td>
<td>1649954</td>
<td>304</td>
<td>9570</td>
</tr>
<tr>
<td>5</td>
<td>475693</td>
<td>472644</td>
<td>1253774</td>
<td>1254138</td>
<td>207</td>
<td>8803</td>
</tr>
<tr>
<td>6</td>
<td>301852</td>
<td>297263</td>
<td>754423</td>
<td>749330</td>
<td>296</td>
<td>7827</td>
</tr>
<tr>
<td>7</td>
<td>349375</td>
<td>346586</td>
<td>928868</td>
<td>927854</td>
<td>286</td>
<td>7161</td>
</tr>
<tr>
<td>8</td>
<td>206792</td>
<td>201194</td>
<td>520408</td>
<td>514621</td>
<td>250</td>
<td>6246</td>
</tr>
<tr>
<td>9</td>
<td>264997</td>
<td>262723</td>
<td>724981</td>
<td>722011</td>
<td>208</td>
<td>5856</td>
</tr>
<tr>
<td>10</td>
<td>138511</td>
<td>136269</td>
<td>341616</td>
<td>341824</td>
<td>191</td>
<td>4405</td>
</tr>
<tr>
<td>11</td>
<td>163585</td>
<td>161860</td>
<td>417785</td>
<td>417034</td>
<td>162</td>
<td>4981</td>
</tr>
<tr>
<td>12</td>
<td>129848</td>
<td>125917</td>
<td>332526</td>
<td>324827</td>
<td>192</td>
<td>4684</td>
</tr>
<tr>
<td>13</td>
<td>238660</td>
<td>237159</td>
<td>593806</td>
<td>587470</td>
<td>186</td>
<td>5361</td>
</tr>
<tr>
<td>14</td>
<td>81506</td>
<td>78375</td>
<td>192482</td>
<td>190105</td>
<td>152</td>
<td>3093</td>
</tr>
<tr>
<td>15</td>
<td>109559</td>
<td>107122</td>
<td>271093</td>
<td>268119</td>
<td>138</td>
<td>3712</td>
</tr>
<tr>
<td>16</td>
<td>58115</td>
<td>56555</td>
<td>138845</td>
<td>138446</td>
<td>90</td>
<td>2921</td>
</tr>
<tr>
<td>17</td>
<td>54268</td>
<td>52268</td>
<td>133802</td>
<td>132536</td>
<td>177</td>
<td>2481</td>
</tr>
<tr>
<td>18</td>
<td>28977</td>
<td>25894</td>
<td>59988</td>
<td>57959</td>
<td>182</td>
<td>3145</td>
</tr>
<tr>
<td>19</td>
<td>12622</td>
<td>10054</td>
<td>24291</td>
<td>21150</td>
<td>151</td>
<td>1898</td>
</tr>
<tr>
<td>20</td>
<td>14756</td>
<td>12133</td>
<td>30541</td>
<td>26655</td>
<td>166</td>
<td>2167</td>
</tr>
<tr>
<td>21</td>
<td>27700</td>
<td>26036</td>
<td>62268</td>
<td>60638</td>
<td>131</td>
<td>1900</td>
</tr>
<tr>
<td>22</td>
<td>8067</td>
<td>7059</td>
<td>16225</td>
<td>14895</td>
<td>96</td>
<td>1010</td>
</tr>
<tr>
<td>23</td>
<td>3952</td>
<td>2962</td>
<td>7254</td>
<td>6505</td>
<td>99</td>
<td>781</td>
</tr>
<tr>
<td>X</td>
<td>378784</td>
<td>377725</td>
<td>1104499</td>
<td>1102248</td>
<td>94</td>
<td>6894</td>
</tr>
<tr>
<td>Y</td>
<td>1474</td>
<td>1502</td>
<td>3881</td>
<td>3876</td>
<td>2</td>
<td>393</td>
</tr>
</tbody>
</table>

Tabelle C.6: Anzahl Duplikate nach Klassen aufgeschlüsselt
### C.2.4  *homo sapiens* (maskiert)

<table>
<thead>
<tr>
<th>Chr.</th>
<th>PID 1</th>
<th>PID 2</th>
<th>PID 3</th>
<th>PID 4</th>
<th>PID 5</th>
<th>PID 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12653</td>
<td>15461</td>
<td>10412</td>
<td>12807</td>
<td>0</td>
<td>3011</td>
</tr>
<tr>
<td>2</td>
<td>6564</td>
<td>7803</td>
<td>11858</td>
<td>0</td>
<td>2455</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>869</td>
<td>1511</td>
<td>1161</td>
<td>0</td>
<td>317</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>2512</td>
<td>3463</td>
<td>1979</td>
<td>0</td>
<td>760</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>6899</td>
<td>9072</td>
<td>8506</td>
<td>6</td>
<td>2039</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>1880</td>
<td>2512</td>
<td>1369</td>
<td>0</td>
<td>892</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7037</td>
<td>8810</td>
<td>11543</td>
<td>3</td>
<td>2216</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>5738</td>
<td>7413</td>
<td>8472</td>
<td>2</td>
<td>755</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>8788</td>
<td>8581</td>
<td>10502</td>
<td>1</td>
<td>3874</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10407</td>
<td>10839</td>
<td>6997</td>
<td>0</td>
<td>3033</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>1851</td>
<td>3180</td>
<td>5719</td>
<td>0</td>
<td>664</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>788</td>
<td>1163</td>
<td>1115</td>
<td>0</td>
<td>357</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>460</td>
<td>539</td>
<td>857</td>
<td>0</td>
<td>249</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>963</td>
<td>1272</td>
<td>265</td>
<td>0</td>
<td>379</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>8640</td>
<td>9968</td>
<td>11498</td>
<td>0</td>
<td>2740</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>14788</td>
<td>15800</td>
<td>14754</td>
<td>0</td>
<td>3796</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>5978</td>
<td>6931</td>
<td>5319</td>
<td>0</td>
<td>1944</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>178</td>
<td>185</td>
<td>240</td>
<td>0</td>
<td>78</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>3661</td>
<td>5397</td>
<td>2549</td>
<td>6</td>
<td>942</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>317</td>
<td>518</td>
<td>416</td>
<td>0</td>
<td>118</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>266</td>
<td>355</td>
<td>86</td>
<td>0</td>
<td>114</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>3071</td>
<td>3491</td>
<td>3853</td>
<td>1</td>
<td>1175</td>
<td></td>
</tr>
<tr>
<td>X</td>
<td>3450</td>
<td>5066</td>
<td>3858</td>
<td>5</td>
<td>999</td>
<td></td>
</tr>
<tr>
<td>Y</td>
<td>4517</td>
<td>4910</td>
<td>5934</td>
<td>0</td>
<td>1642</td>
<td></td>
</tr>
</tbody>
</table>

Tabelle C.7: Anzahl Duplikate nach Klassen aufgeschlüsselt
### C.3 Anzahl redundanter und nichtredundanter Repeats und Palindrome

#### C.3.1 *homo sapiens* (unmaskiert)

<table>
<thead>
<tr>
<th>Chr.</th>
<th>Tupel insg</th>
<th>Tupel red</th>
<th>Tupel nichtred</th>
<th>Red. in Tupel insg</th>
<th>Tupel red</th>
<th>Tupel nichtred</th>
<th>Red. in Tupel insg</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1406647</td>
<td>12760</td>
<td>1393887</td>
<td>0.90712</td>
<td>3737010</td>
<td>294556</td>
<td>3442454</td>
</tr>
<tr>
<td>2</td>
<td>1794139</td>
<td>15292</td>
<td>1609235</td>
<td>0.85233</td>
<td>4754577</td>
<td>386158</td>
<td>4368419</td>
</tr>
<tr>
<td>3</td>
<td>1622468</td>
<td>13233</td>
<td>1609235</td>
<td>0.81561</td>
<td>5118854</td>
<td>384562</td>
<td>4734292</td>
</tr>
<tr>
<td>4</td>
<td>1836311</td>
<td>14958</td>
<td>1821353</td>
<td>0.81457</td>
<td>5118854</td>
<td>384562</td>
<td>4734292</td>
</tr>
<tr>
<td>5</td>
<td>1512938</td>
<td>12277</td>
<td>1500661</td>
<td>0.81147</td>
<td>5118854</td>
<td>384562</td>
<td>4734292</td>
</tr>
<tr>
<td>6</td>
<td>1076594</td>
<td>8759</td>
<td>1067835</td>
<td>0.81358</td>
<td>3036792</td>
<td>221802</td>
<td>2814990</td>
</tr>
<tr>
<td>7</td>
<td>776815</td>
<td>7911</td>
<td>768904</td>
<td>0.81001</td>
<td>221802</td>
<td>160743</td>
<td>1855974</td>
</tr>
<tr>
<td>8</td>
<td>830854</td>
<td>6730</td>
<td>824124</td>
<td>0.81839</td>
<td>221802</td>
<td>160743</td>
<td>1855974</td>
</tr>
<tr>
<td>9</td>
<td>459245</td>
<td>4389</td>
<td>454856</td>
<td>0.81457</td>
<td>221802</td>
<td>160743</td>
<td>1855974</td>
</tr>
<tr>
<td>10</td>
<td>494206</td>
<td>4714</td>
<td>489492</td>
<td>0.81001</td>
<td>221802</td>
<td>160743</td>
<td>1855974</td>
</tr>
<tr>
<td>11</td>
<td>731014</td>
<td>5825</td>
<td>725189</td>
<td>0.79684</td>
<td>2016717</td>
<td>160743</td>
<td>1855974</td>
</tr>
<tr>
<td>12</td>
<td>574496</td>
<td>4757</td>
<td>569739</td>
<td>0.82803</td>
<td>1561314</td>
<td>115544</td>
<td>1445770</td>
</tr>
<tr>
<td>13</td>
<td>252882</td>
<td>2374</td>
<td>250508</td>
<td>0.95385</td>
<td>1561314</td>
<td>115544</td>
<td>1445770</td>
</tr>
<tr>
<td>14</td>
<td>260585</td>
<td>2268</td>
<td>258317</td>
<td>0.95385</td>
<td>1561314</td>
<td>115544</td>
<td>1445770</td>
</tr>
<tr>
<td>15</td>
<td>172634</td>
<td>2031</td>
<td>170603</td>
<td>1.17648</td>
<td>1561314</td>
<td>115544</td>
<td>1445770</td>
</tr>
<tr>
<td>16</td>
<td>119096</td>
<td>2426</td>
<td>116670</td>
<td>2.03701</td>
<td>1561314</td>
<td>115544</td>
<td>1445770</td>
</tr>
<tr>
<td>17</td>
<td>70715</td>
<td>1102</td>
<td>69613</td>
<td>1.55837</td>
<td>1561314</td>
<td>115544</td>
<td>1445770</td>
</tr>
<tr>
<td>18</td>
<td>163496</td>
<td>1637</td>
<td>161859</td>
<td>1.00125</td>
<td>1561314</td>
<td>115544</td>
<td>1445770</td>
</tr>
<tr>
<td>19</td>
<td>46717</td>
<td>641</td>
<td>46076</td>
<td>1.37209</td>
<td>1561314</td>
<td>115544</td>
<td>1445770</td>
</tr>
<tr>
<td>20</td>
<td>56526</td>
<td>577</td>
<td>57049</td>
<td>0.95171</td>
<td>1561314</td>
<td>115544</td>
<td>1445770</td>
</tr>
<tr>
<td>21</td>
<td>17052</td>
<td>165</td>
<td>16887</td>
<td>0.96763</td>
<td>1561314</td>
<td>115544</td>
<td>1445770</td>
</tr>
<tr>
<td>22</td>
<td>13226</td>
<td>323</td>
<td>12903</td>
<td>2.44216</td>
<td>1561314</td>
<td>115544</td>
<td>1445770</td>
</tr>
<tr>
<td>X</td>
<td>2272275</td>
<td>15508</td>
<td>2256767</td>
<td>0.68249</td>
<td>6604576</td>
<td>465604</td>
<td>6147972</td>
</tr>
<tr>
<td>Y</td>
<td>56778</td>
<td>874</td>
<td>59004</td>
<td>1.53933</td>
<td>160273</td>
<td>9408</td>
<td>150865</td>
</tr>
<tr>
<td>∑</td>
<td>16619709</td>
<td>141511</td>
<td>16478198</td>
<td>∅ 0.85146</td>
<td>45462559</td>
<td>3442282</td>
<td>42020277</td>
</tr>
</tbody>
</table>

Tabelle C.8: Anzahl redundanter und nichtredundanter Repeats
ANHANG C. TABELLEN MIT ANZAHL DER DUPLIKATE

C.3.2 *pan troglodytes* (unmaskiert)

<table>
<thead>
<tr>
<th>Chr.</th>
<th>Tupel insg.</th>
<th>red. Tupel</th>
<th>nichtred. Tupel</th>
<th>Red. in Prozent</th>
<th>Tupel insg.</th>
<th>red. Tupel</th>
<th>nichtred. Tupel</th>
<th>Red. in Prozent</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>500335</td>
<td>3566</td>
<td>496769</td>
<td>0.712722</td>
<td>1299952</td>
<td>88965</td>
<td>1210987</td>
<td>6.84371</td>
</tr>
<tr>
<td>2</td>
<td>676325</td>
<td>3973</td>
<td>672352</td>
<td>0.587439</td>
<td>1789257</td>
<td>123911</td>
<td>1665346</td>
<td>6.92528</td>
</tr>
<tr>
<td>3</td>
<td>770983</td>
<td>4936</td>
<td>76047</td>
<td>0.640222</td>
<td>2065651</td>
<td>144949</td>
<td>1920702</td>
<td>7.01711</td>
</tr>
<tr>
<td>4</td>
<td>631650</td>
<td>3735</td>
<td>627915</td>
<td>0.591308</td>
<td>1653232</td>
<td>115600</td>
<td>1537632</td>
<td>6.99236</td>
</tr>
<tr>
<td>5</td>
<td>475692</td>
<td>2932</td>
<td>472760</td>
<td>0.616365</td>
<td>1253773</td>
<td>85534</td>
<td>1168239</td>
<td>6.82236</td>
</tr>
<tr>
<td>6</td>
<td>301851</td>
<td>2537</td>
<td>299314</td>
<td>0.640222</td>
<td>2065651</td>
<td>144949</td>
<td>1920702</td>
<td>7.01711</td>
</tr>
<tr>
<td>7</td>
<td>264996</td>
<td>1473</td>
<td>263523</td>
<td>0.555857</td>
<td>1253773</td>
<td>85534</td>
<td>1168239</td>
<td>6.29631</td>
</tr>
<tr>
<td>8</td>
<td>349374</td>
<td>1660</td>
<td>346951</td>
<td>0.695555</td>
<td>593805</td>
<td>46533</td>
<td>547272</td>
<td>7.83641</td>
</tr>
<tr>
<td>9</td>
<td>206791</td>
<td>1465</td>
<td>205326</td>
<td>0.708445</td>
<td>520407</td>
<td>37172</td>
<td>483235</td>
<td>7.14287</td>
</tr>
<tr>
<td>10</td>
<td>138510</td>
<td>965</td>
<td>137545</td>
<td>0.696701</td>
<td>341615</td>
<td>25294</td>
<td>316321</td>
<td>7.40424</td>
</tr>
<tr>
<td>11</td>
<td>163584</td>
<td>1123</td>
<td>162461</td>
<td>0.68649</td>
<td>417784</td>
<td>30386</td>
<td>387398</td>
<td>7.27314</td>
</tr>
<tr>
<td>12</td>
<td>129847</td>
<td>973</td>
<td>128874</td>
<td>0.749343</td>
<td>332525</td>
<td>24344</td>
<td>308181</td>
<td>7.32095</td>
</tr>
<tr>
<td>13</td>
<td>238659</td>
<td>1660</td>
<td>236999</td>
<td>0.695553</td>
<td>593805</td>
<td>46533</td>
<td>547272</td>
<td>7.83641</td>
</tr>
<tr>
<td>14</td>
<td>81505</td>
<td>698</td>
<td>80807</td>
<td>0.856389</td>
<td>192481</td>
<td>15839</td>
<td>176642</td>
<td>8.22886</td>
</tr>
<tr>
<td>15</td>
<td>109558</td>
<td>778</td>
<td>108780</td>
<td>0.710126</td>
<td>271092</td>
<td>19970</td>
<td>251122</td>
<td>7.3665</td>
</tr>
<tr>
<td>16</td>
<td>58114</td>
<td>508</td>
<td>57606</td>
<td>0.874144</td>
<td>138844</td>
<td>10962</td>
<td>127882</td>
<td>7.89519</td>
</tr>
<tr>
<td>17</td>
<td>54267</td>
<td>373</td>
<td>53894</td>
<td>0.687342</td>
<td>133801</td>
<td>9724</td>
<td>124077</td>
<td>7.26751</td>
</tr>
<tr>
<td>18</td>
<td>28976</td>
<td>354</td>
<td>28622</td>
<td>1.2217</td>
<td>59987</td>
<td>5234</td>
<td>54753</td>
<td>8.72522</td>
</tr>
<tr>
<td>19</td>
<td>12621</td>
<td>194</td>
<td>12427</td>
<td>1.53712</td>
<td>24290</td>
<td>1772</td>
<td>22518</td>
<td>7.29518</td>
</tr>
<tr>
<td>20</td>
<td>14755</td>
<td>123</td>
<td>14632</td>
<td>0.833616</td>
<td>30540</td>
<td>1580</td>
<td>28960</td>
<td>5.17354</td>
</tr>
<tr>
<td>21</td>
<td>27699</td>
<td>281</td>
<td>27418</td>
<td>1.01448</td>
<td>62267</td>
<td>5010</td>
<td>57257</td>
<td>8.046</td>
</tr>
<tr>
<td>22</td>
<td>8066</td>
<td>95</td>
<td>7971</td>
<td>1.17778</td>
<td>16224</td>
<td>1312</td>
<td>14912</td>
<td>8.08679</td>
</tr>
<tr>
<td>23</td>
<td>3951</td>
<td>57</td>
<td>3894</td>
<td>1.44267</td>
<td>7253</td>
<td>462</td>
<td>6791</td>
<td>6.36978</td>
</tr>
<tr>
<td>X</td>
<td>378783</td>
<td>1666</td>
<td>377117</td>
<td>0.43983</td>
<td>1104498</td>
<td>63430</td>
<td>1041068</td>
<td>5.74288</td>
</tr>
<tr>
<td>Y</td>
<td>1473</td>
<td>29</td>
<td>1444</td>
<td>1.96877</td>
<td>3880</td>
<td>213</td>
<td>3667</td>
<td>5.49609</td>
</tr>
<tr>
<td>∑</td>
<td>5628365</td>
<td>36917</td>
<td>5591448</td>
<td>0.65590</td>
<td>14721427</td>
<td>1032538</td>
<td>13688889</td>
<td>7.01384</td>
</tr>
</tbody>
</table>

Tabelle C.9: Anzahl redundanter und nichtredundanter Repeats
### C.3.3 *homo sapiens* (maskiert)

<table>
<thead>
<tr>
<th>Chr.</th>
<th>Tupel insg.</th>
<th>red. Tupel</th>
<th>nichtred. Tupel</th>
<th>Red. in %</th>
<th>Tupel insg.</th>
<th>red. Tupel</th>
<th>nichtred. Tupel</th>
<th>Red. in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>12652</td>
<td>681</td>
<td>11971</td>
<td>5.38255</td>
<td>15460</td>
<td>3500</td>
<td>11960</td>
<td>22.6391</td>
</tr>
<tr>
<td>2</td>
<td>6563</td>
<td>328</td>
<td>6235</td>
<td>4.99771</td>
<td>7802</td>
<td>1983</td>
<td>5819</td>
<td>25.4166</td>
</tr>
<tr>
<td>3</td>
<td>868</td>
<td>17</td>
<td>851</td>
<td>1.95853</td>
<td>1510</td>
<td>498</td>
<td>11960</td>
<td>13.7086</td>
</tr>
<tr>
<td>4</td>
<td>2511</td>
<td>94</td>
<td>2417</td>
<td>3.74353</td>
<td>4160</td>
<td>1236</td>
<td>2924</td>
<td>19.8327</td>
</tr>
<tr>
<td>5</td>
<td>6898</td>
<td>313</td>
<td>6585</td>
<td>4.53755</td>
<td>9071</td>
<td>1519</td>
<td>7552</td>
<td>16.7457</td>
</tr>
<tr>
<td>6</td>
<td>1879</td>
<td>69</td>
<td>1810</td>
<td>3.67217</td>
<td>2511</td>
<td>746</td>
<td>2716</td>
<td>21.5482</td>
</tr>
<tr>
<td>7</td>
<td>7036</td>
<td>370</td>
<td>6666</td>
<td>5.25867</td>
<td>9802</td>
<td>2409</td>
<td>7393</td>
<td>22.3635</td>
</tr>
<tr>
<td>8</td>
<td>5737</td>
<td>265</td>
<td>5472</td>
<td>4.61914</td>
<td>7412</td>
<td>1988</td>
<td>5424</td>
<td>20.0756</td>
</tr>
<tr>
<td>9</td>
<td>8787</td>
<td>676</td>
<td>8111</td>
<td>7.69318</td>
<td>10838</td>
<td>3000</td>
<td>7838</td>
<td>27.6804</td>
</tr>
<tr>
<td>10</td>
<td>10406</td>
<td>723</td>
<td>9683</td>
<td>6.94791</td>
<td>10838</td>
<td>3000</td>
<td>7838</td>
<td>27.6804</td>
</tr>
<tr>
<td>11</td>
<td>1850</td>
<td>45</td>
<td>1805</td>
<td>2.43243</td>
<td>3179</td>
<td>439</td>
<td>2740</td>
<td>13.8094</td>
</tr>
<tr>
<td>12</td>
<td>787</td>
<td>22</td>
<td>765</td>
<td>2.79543</td>
<td>1162</td>
<td>242</td>
<td>920</td>
<td>20.8262</td>
</tr>
<tr>
<td>13</td>
<td>459</td>
<td>8</td>
<td>451</td>
<td>1.74292</td>
<td>538</td>
<td>103</td>
<td>435</td>
<td>19.145</td>
</tr>
<tr>
<td>14</td>
<td>962</td>
<td>37</td>
<td>925</td>
<td>3.84615</td>
<td>1271</td>
<td>262</td>
<td>1009</td>
<td>20.6137</td>
</tr>
<tr>
<td>15</td>
<td>8639</td>
<td>595</td>
<td>8044</td>
<td>6.88737</td>
<td>9967</td>
<td>2269</td>
<td>7698</td>
<td>22.7651</td>
</tr>
<tr>
<td>16</td>
<td>14787</td>
<td>918</td>
<td>13869</td>
<td>6.20816</td>
<td>15799</td>
<td>4871</td>
<td>10928</td>
<td>30.8311</td>
</tr>
<tr>
<td>17</td>
<td>5977</td>
<td>376</td>
<td>5601</td>
<td>6.29078</td>
<td>6930</td>
<td>1952</td>
<td>4978</td>
<td>28.1674</td>
</tr>
<tr>
<td>18</td>
<td>177</td>
<td>9</td>
<td>168</td>
<td>5.08475</td>
<td>184</td>
<td>35</td>
<td>149</td>
<td>19.0217</td>
</tr>
<tr>
<td>19</td>
<td>3660</td>
<td>144</td>
<td>3516</td>
<td>3.93443</td>
<td>5396</td>
<td>690</td>
<td>4766</td>
<td>12.7872</td>
</tr>
<tr>
<td>20</td>
<td>316</td>
<td>6</td>
<td>310</td>
<td>1.89873</td>
<td>517</td>
<td>81</td>
<td>436</td>
<td>15.6673</td>
</tr>
<tr>
<td>21</td>
<td>265</td>
<td>6</td>
<td>259</td>
<td>2.26415</td>
<td>354</td>
<td>93</td>
<td>261</td>
<td>26.2712</td>
</tr>
<tr>
<td>22</td>
<td>3070</td>
<td>194</td>
<td>2876</td>
<td>6.31922</td>
<td>3490</td>
<td>880</td>
<td>2610</td>
<td>25.2149</td>
</tr>
<tr>
<td>X</td>
<td>3449</td>
<td>230</td>
<td>3219</td>
<td>6.6686</td>
<td>5065</td>
<td>801</td>
<td>4264</td>
<td>15.8144</td>
</tr>
<tr>
<td>Y</td>
<td>4516</td>
<td>377</td>
<td>4139</td>
<td>8.3481</td>
<td>4909</td>
<td>765</td>
<td>4144</td>
<td>15.5836</td>
</tr>
<tr>
<td>∑</td>
<td>112251</td>
<td>6503</td>
<td>105748</td>
<td>5,79326</td>
<td>134216</td>
<td>30803</td>
<td>103413</td>
<td>22.95031</td>
</tr>
</tbody>
</table>

Tabelle C.10: Anzahl redundanter und nichtredundanter Repeats
### C.3.4 *homo sapiens* (unmaskiert)

<table>
<thead>
<tr>
<th>Chr.</th>
<th>Tupel insg.</th>
<th>red. Tupel</th>
<th>nichtred. Tupel</th>
<th>Red. in %</th>
<th>Tupel insg.</th>
<th>red. Tupel</th>
<th>nichtred. Tupel</th>
<th>Red. in %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1399573</td>
<td>12647</td>
<td>1386926</td>
<td>0.903633</td>
<td>3727198</td>
<td>288906</td>
<td>3438292</td>
<td>7.75129</td>
</tr>
<tr>
<td>2</td>
<td>1792342</td>
<td>16152</td>
<td>1776190</td>
<td>0.901167</td>
<td>4737182</td>
<td>386080</td>
<td>4113404</td>
<td>7.46029</td>
</tr>
<tr>
<td>3</td>
<td>1618999</td>
<td>13264</td>
<td>1605735</td>
<td>0.819272</td>
<td>3005440</td>
<td>221493</td>
<td>2783947</td>
<td>7.36974</td>
</tr>
<tr>
<td>4</td>
<td>1829664</td>
<td>14717</td>
<td>1814947</td>
<td>0.804355</td>
<td>4351102</td>
<td>363397</td>
<td>3928973</td>
<td>7.18597</td>
</tr>
<tr>
<td>5</td>
<td>1510723</td>
<td>12273</td>
<td>1498450</td>
<td>0.812392</td>
<td>4235226</td>
<td>331611</td>
<td>3928973</td>
<td>7.18597</td>
</tr>
<tr>
<td>6</td>
<td>1067843</td>
<td>8769</td>
<td>1059074</td>
<td>0.821188</td>
<td>288906</td>
<td>2106768</td>
<td>2106768</td>
<td>7.52716</td>
</tr>
<tr>
<td>7</td>
<td>759466</td>
<td>7318</td>
<td>752148</td>
<td>0.863572</td>
<td>1992035</td>
<td>161547</td>
<td>1830471</td>
<td>7.8844</td>
</tr>
<tr>
<td>8</td>
<td>827741</td>
<td>6847</td>
<td>820894</td>
<td>0.827191</td>
<td>2278256</td>
<td>171488</td>
<td>2106768</td>
<td>7.52716</td>
</tr>
<tr>
<td>9</td>
<td>462151</td>
<td>4545</td>
<td>457606</td>
<td>0.983445</td>
<td>1305471</td>
<td>100316</td>
<td>1205155</td>
<td>7.31849</td>
</tr>
<tr>
<td>10</td>
<td>478012</td>
<td>4260</td>
<td>473752</td>
<td>0.891191</td>
<td>1305471</td>
<td>100316</td>
<td>1205155</td>
<td>7.31849</td>
</tr>
<tr>
<td>11</td>
<td>732450</td>
<td>5758</td>
<td>726692</td>
<td>0.786129</td>
<td>2061271</td>
<td>150854</td>
<td>1910417</td>
<td>7.31849</td>
</tr>
<tr>
<td>12</td>
<td>568027</td>
<td>4779</td>
<td>563248</td>
<td>0.841333</td>
<td>1555358</td>
<td>117138</td>
<td>1438220</td>
<td>7.53126</td>
</tr>
<tr>
<td>13</td>
<td>248338</td>
<td>2541</td>
<td>245797</td>
<td>1.0232</td>
<td>626926</td>
<td>55611</td>
<td>571315</td>
<td>8.87042</td>
</tr>
<tr>
<td>14</td>
<td>252982</td>
<td>2106</td>
<td>250876</td>
<td>0.83247</td>
<td>682899</td>
<td>54762</td>
<td>628137</td>
<td>8.01905</td>
</tr>
<tr>
<td>15</td>
<td>170644</td>
<td>2128</td>
<td>168516</td>
<td>1.24704</td>
<td>427548</td>
<td>36787</td>
<td>390761</td>
<td>8.60418</td>
</tr>
<tr>
<td>16</td>
<td>13738</td>
<td>2283</td>
<td>111455</td>
<td>2.00724</td>
<td>251123</td>
<td>26674</td>
<td>224449</td>
<td>10.6219</td>
</tr>
<tr>
<td>17</td>
<td>64149</td>
<td>1000</td>
<td>63149</td>
<td>1.55887</td>
<td>148676</td>
<td>11829</td>
<td>136847</td>
<td>7.95623</td>
</tr>
<tr>
<td>18</td>
<td>161861</td>
<td>1608</td>
<td>160253</td>
<td>0.993445</td>
<td>415719</td>
<td>35610</td>
<td>380160</td>
<td>8.56588</td>
</tr>
<tr>
<td>19</td>
<td>41840</td>
<td>387</td>
<td>41093</td>
<td>0.930298</td>
<td>96485</td>
<td>6691</td>
<td>89794</td>
<td>6.93476</td>
</tr>
<tr>
<td>20</td>
<td>56761</td>
<td>513</td>
<td>56248</td>
<td>0.90379</td>
<td>141188</td>
<td>12599</td>
<td>128589</td>
<td>8.92356</td>
</tr>
<tr>
<td>21</td>
<td>15276</td>
<td>128</td>
<td>15148</td>
<td>0.837916</td>
<td>36380</td>
<td>3110</td>
<td>33070</td>
<td>9.09841</td>
</tr>
<tr>
<td>22</td>
<td>12427</td>
<td>305</td>
<td>12122</td>
<td>2.45433</td>
<td>23730</td>
<td>2413</td>
<td>21317</td>
<td>10.1686</td>
</tr>
<tr>
<td>X</td>
<td>2259751</td>
<td>15240</td>
<td>2244511</td>
<td>0.674411</td>
<td>6587645</td>
<td>451803</td>
<td>6135843</td>
<td>6.85834</td>
</tr>
<tr>
<td>Y</td>
<td>45131</td>
<td>476</td>
<td>44655</td>
<td>1.05471</td>
<td>136522</td>
<td>7700</td>
<td>128822</td>
<td>5.64012</td>
</tr>
<tr>
<td>∑</td>
<td>16489529</td>
<td>140044</td>
<td>16349485</td>
<td>0.84929</td>
<td>45270907</td>
<td>3418580</td>
<td>41852327</td>
<td>7.55138</td>
</tr>
</tbody>
</table>

Tabelle C.11: Anzahl redundanter und nichtredundanter Palindrome
### C.3.5 *pan troglodytes* (unmaskiert)

<table>
<thead>
<tr>
<th>Chr.</th>
<th>Tupel insg.</th>
<th>red. Tupel</th>
<th>nichtred. Tupel</th>
<th>Red. in Tupel</th>
<th>Tupel insg.</th>
<th>red. Tupel</th>
<th>nichtred. Tupel</th>
<th>Red. in Tupel</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>496225</td>
<td>3214</td>
<td>493011</td>
<td>0.64769</td>
<td>1301560</td>
<td>88560</td>
<td>1213000</td>
<td>0.680414</td>
</tr>
<tr>
<td>2</td>
<td>672560</td>
<td>3631</td>
<td>668929</td>
<td>0.539877</td>
<td>1788116</td>
<td>124272</td>
<td>1663844</td>
<td>0.94988</td>
</tr>
<tr>
<td>3</td>
<td>767536</td>
<td>4543</td>
<td>762993</td>
<td>0.591894</td>
<td>1649953</td>
<td>114306</td>
<td>1535647</td>
<td>0.92783</td>
</tr>
<tr>
<td>4</td>
<td>627159</td>
<td>3602</td>
<td>623557</td>
<td>0.574336</td>
<td>1649953</td>
<td>114306</td>
<td>1535647</td>
<td>0.92783</td>
</tr>
<tr>
<td>5</td>
<td>472643</td>
<td>2924</td>
<td>469719</td>
<td>0.618649</td>
<td>1254137</td>
<td>85240</td>
<td>1168897</td>
<td>0.79671</td>
</tr>
<tr>
<td>6</td>
<td>297262</td>
<td>2214</td>
<td>295048</td>
<td>0.744798</td>
<td>749329</td>
<td>58835</td>
<td>690494</td>
<td>0.78516</td>
</tr>
<tr>
<td>7</td>
<td>346585</td>
<td>2361</td>
<td>344224</td>
<td>0.68394</td>
<td>514620</td>
<td>36079</td>
<td>478541</td>
<td>0.70108</td>
</tr>
<tr>
<td>8</td>
<td>201193</td>
<td>1385</td>
<td>199808</td>
<td>0.688394</td>
<td>927853</td>
<td>69026</td>
<td>858827</td>
<td>0.743932</td>
</tr>
<tr>
<td>9</td>
<td>262722</td>
<td>1474</td>
<td>261248</td>
<td>0.561049</td>
<td>722010</td>
<td>46580</td>
<td>675430</td>
<td>0.645143</td>
</tr>
<tr>
<td>10</td>
<td>136268</td>
<td>845</td>
<td>135423</td>
<td>0.620102</td>
<td>341823</td>
<td>25572</td>
<td>316251</td>
<td>0.748106</td>
</tr>
<tr>
<td>11</td>
<td>161859</td>
<td>1054</td>
<td>160805</td>
<td>0.651184</td>
<td>417033</td>
<td>29137</td>
<td>387896</td>
<td>0.698674</td>
</tr>
<tr>
<td>12</td>
<td>125916</td>
<td>887</td>
<td>125029</td>
<td>0.704438</td>
<td>324826</td>
<td>23548</td>
<td>301278</td>
<td>0.724942</td>
</tr>
<tr>
<td>13</td>
<td>237158</td>
<td>1645</td>
<td>235513</td>
<td>0.69363</td>
<td>587469</td>
<td>42048</td>
<td>541261</td>
<td>0.785661</td>
</tr>
<tr>
<td>14</td>
<td>79374</td>
<td>555</td>
<td>78819</td>
<td>0.699221</td>
<td>190104</td>
<td>15458</td>
<td>174646</td>
<td>0.813134</td>
</tr>
<tr>
<td>15</td>
<td>107121</td>
<td>676</td>
<td>106445</td>
<td>0.631062</td>
<td>268118</td>
<td>19736</td>
<td>248382</td>
<td>0.736094</td>
</tr>
<tr>
<td>16</td>
<td>56554</td>
<td>376</td>
<td>56178</td>
<td>0.664851</td>
<td>138445</td>
<td>10365</td>
<td>128080</td>
<td>0.748673</td>
</tr>
<tr>
<td>17</td>
<td>52267</td>
<td>356</td>
<td>51911</td>
<td>0.681118</td>
<td>132535</td>
<td>9639</td>
<td>122896</td>
<td>0.72728</td>
</tr>
<tr>
<td>18</td>
<td>25893</td>
<td>249</td>
<td>25644</td>
<td>0.96165</td>
<td>57958</td>
<td>4768</td>
<td>53190</td>
<td>0.822665</td>
</tr>
<tr>
<td>19</td>
<td>10053</td>
<td>119</td>
<td>9934</td>
<td>1.18373</td>
<td>21149</td>
<td>1486</td>
<td>19663</td>
<td>0.762634</td>
</tr>
<tr>
<td>20</td>
<td>12132</td>
<td>66</td>
<td>12066</td>
<td>0.544016</td>
<td>26654</td>
<td>1222</td>
<td>25432</td>
<td>0.548468</td>
</tr>
<tr>
<td>21</td>
<td>26035</td>
<td>185</td>
<td>25850</td>
<td>0.710582</td>
<td>60637</td>
<td>4805</td>
<td>55832</td>
<td>0.79242</td>
</tr>
<tr>
<td>22</td>
<td>7058</td>
<td>56</td>
<td>7002</td>
<td>0.793426</td>
<td>14894</td>
<td>1168</td>
<td>13726</td>
<td>0.784208</td>
</tr>
<tr>
<td>23</td>
<td>2961</td>
<td>45</td>
<td>2916</td>
<td>1.51976</td>
<td>6504</td>
<td>415</td>
<td>6089</td>
<td>0.638069</td>
</tr>
<tr>
<td>X</td>
<td>377724</td>
<td>1594</td>
<td>376130</td>
<td>0.422001</td>
<td>1102247</td>
<td>63423</td>
<td>1038824</td>
<td>0.57397</td>
</tr>
<tr>
<td>Y</td>
<td>1501</td>
<td>23</td>
<td>1478</td>
<td>1.53231</td>
<td>3875</td>
<td>227</td>
<td>3648</td>
<td>0.585866</td>
</tr>
<tr>
<td>∑</td>
<td>5563759</td>
<td>34079</td>
<td>5529680</td>
<td>0.61251</td>
<td>1466885</td>
<td>1025320</td>
<td>13643537</td>
<td>0.98977</td>
</tr>
</tbody>
</table>

Tabelle C.12: Anzahl redundanter und nichtredundanter Palindrome
### C.3.6 *homo sapiens* (maskiert)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10411</td>
<td>554</td>
<td>9857</td>
<td>5.3219</td>
<td>12806</td>
<td>3053</td>
<td>9753</td>
<td>23.8404</td>
</tr>
<tr>
<td>2</td>
<td>10849</td>
<td>652</td>
<td>10197</td>
<td>6.0097</td>
<td>11857</td>
<td>3271</td>
<td>8586</td>
<td>27.5871</td>
</tr>
<tr>
<td>3</td>
<td>720</td>
<td>19</td>
<td>701</td>
<td>2.6388</td>
<td>1160</td>
<td>145</td>
<td>1015</td>
<td>12.5</td>
</tr>
<tr>
<td>4</td>
<td>1182</td>
<td>40</td>
<td>1142</td>
<td>3.3840</td>
<td>1978</td>
<td>312</td>
<td>1666</td>
<td>15.7735</td>
</tr>
<tr>
<td>5</td>
<td>7297</td>
<td>422</td>
<td>6875</td>
<td>5.7832</td>
<td>8505</td>
<td>1956</td>
<td>6549</td>
<td>22.9982</td>
</tr>
<tr>
<td>6</td>
<td>991</td>
<td>48</td>
<td>943</td>
<td>4.8435</td>
<td>1368</td>
<td>278</td>
<td>1090</td>
<td>20.3216</td>
</tr>
<tr>
<td>7</td>
<td>10354</td>
<td>625</td>
<td>9729</td>
<td>6.0363</td>
<td>11542</td>
<td>2907</td>
<td>8635</td>
<td>25.1863</td>
</tr>
<tr>
<td>8</td>
<td>6433</td>
<td>380</td>
<td>6053</td>
<td>5.9070</td>
<td>8471</td>
<td>1801</td>
<td>6670</td>
<td>21.2608</td>
</tr>
<tr>
<td>9</td>
<td>10835</td>
<td>792</td>
<td>10043</td>
<td>7.3096</td>
<td>10501</td>
<td>3318</td>
<td>7183</td>
<td>31.597</td>
</tr>
<tr>
<td>10</td>
<td>6291</td>
<td>338</td>
<td>5953</td>
<td>5.3727</td>
<td>6996</td>
<td>1905</td>
<td>5091</td>
<td>27.2298</td>
</tr>
<tr>
<td>11</td>
<td>3675</td>
<td>107</td>
<td>3568</td>
<td>2.9115</td>
<td>5718</td>
<td>1093</td>
<td>4625</td>
<td>19.1151</td>
</tr>
<tr>
<td>12</td>
<td>984</td>
<td>45</td>
<td>939</td>
<td>4.5731</td>
<td>1114</td>
<td>336</td>
<td>778</td>
<td>30.1616</td>
</tr>
<tr>
<td>13</td>
<td>712</td>
<td>37</td>
<td>675</td>
<td>5.1966</td>
<td>856</td>
<td>214</td>
<td>642</td>
<td>25</td>
</tr>
<tr>
<td>14</td>
<td>185</td>
<td>7</td>
<td>178</td>
<td>3.7837</td>
<td>264</td>
<td>43</td>
<td>221</td>
<td>16.2879</td>
</tr>
<tr>
<td>15</td>
<td>10389</td>
<td>792</td>
<td>9597</td>
<td>7.6234</td>
<td>11497</td>
<td>2827</td>
<td>8670</td>
<td>24.589</td>
</tr>
<tr>
<td>16</td>
<td>14467</td>
<td>999</td>
<td>13468</td>
<td>6.9053</td>
<td>14753</td>
<td>4545</td>
<td>10208</td>
<td>30.8073</td>
</tr>
<tr>
<td>17</td>
<td>4777</td>
<td>294</td>
<td>4483</td>
<td>6.1544</td>
<td>5318</td>
<td>1236</td>
<td>4082</td>
<td>23.2418</td>
</tr>
<tr>
<td>18</td>
<td>232</td>
<td>9</td>
<td>223</td>
<td>3.8793</td>
<td>239</td>
<td>74</td>
<td>165</td>
<td>30.9623</td>
</tr>
<tr>
<td>19</td>
<td>2444</td>
<td>96</td>
<td>2348</td>
<td>3.9279</td>
<td>2548</td>
<td>540</td>
<td>2008</td>
<td>21.1931</td>
</tr>
<tr>
<td>20</td>
<td>366</td>
<td>18</td>
<td>348</td>
<td>4.9180</td>
<td>415</td>
<td>88</td>
<td>327</td>
<td>21.2048</td>
</tr>
<tr>
<td>21</td>
<td>64</td>
<td>1</td>
<td>63</td>
<td>1.5625</td>
<td>85</td>
<td>13</td>
<td>72</td>
<td>15.2941</td>
</tr>
<tr>
<td>22</td>
<td>3345</td>
<td>215</td>
<td>3130</td>
<td>6.4275</td>
<td>3852</td>
<td>1002</td>
<td>2850</td>
<td>26.0125</td>
</tr>
<tr>
<td>X</td>
<td>2917</td>
<td>144</td>
<td>2773</td>
<td>4.9365</td>
<td>3857</td>
<td>477</td>
<td>3380</td>
<td>12.3671</td>
</tr>
<tr>
<td>Y</td>
<td>6059</td>
<td>373</td>
<td>5686</td>
<td>6.1561</td>
<td>5933</td>
<td>764</td>
<td>5169</td>
<td>12.8771</td>
</tr>
</tbody>
</table>

| ∑    | 115979      | 7007       | 108972          | 6.04161             | 131633      | 32198      | 99435           | 24.46043       |

Tabelle C.13: Anzahl redundanter und nichtredundanter Palindrome

### C.4 native Clustering
### Tabelle C.14: Repeatstatistik zum native clustering von *homo sapiens* (unmaskiert)

<table>
<thead>
<tr>
<th>Chr.</th>
<th>PID</th>
<th>NUM_CLUSTERS</th>
<th>NUM_CL_PARTS</th>
<th>MAX_CL_PARTS</th>
<th>MAX_CL_LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>116005</td>
<td>261548</td>
<td>17</td>
<td>40966</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>678227</td>
<td>2324695</td>
<td>58</td>
<td>42842</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>154057</td>
<td>344257</td>
<td>15</td>
<td>11434</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>866256</td>
<td>3024650</td>
<td>57</td>
<td>15742</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>132949</td>
<td>295457</td>
<td>14</td>
<td>3508</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>821889</td>
<td>2869670</td>
<td>57</td>
<td>6044</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>149342</td>
<td>332470</td>
<td>17</td>
<td>20367</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>966592</td>
<td>3323532</td>
<td>60</td>
<td>23748</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>119337</td>
<td>260250</td>
<td>23</td>
<td>20959</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>790372</td>
<td>2707435</td>
<td>171</td>
<td>55623</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>84781</td>
<td>186956</td>
<td>14</td>
<td>13244</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>572292</td>
<td>1939785</td>
<td>59</td>
<td>13317</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>68916</td>
<td>157556</td>
<td>21</td>
<td>14576</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>361978</td>
<td>1273966</td>
<td>123</td>
<td>18715</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>67121</td>
<td>149180</td>
<td>19</td>
<td>27774</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>430127</td>
<td>1483794</td>
<td>72</td>
<td>33727</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>37241</td>
<td>85195</td>
<td>20</td>
<td>16879</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>230010</td>
<td>768360</td>
<td>51</td>
<td>16927</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>42443</td>
<td>96136</td>
<td>21</td>
<td>43394</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>244508</td>
<td>846808</td>
<td>49</td>
<td>43413</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>57684</td>
<td>127032</td>
<td>14</td>
<td>8444</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>388777</td>
<td>1302847</td>
<td>52</td>
<td>13826</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>45889</td>
<td>101561</td>
<td>15</td>
<td>4291</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>283421</td>
<td>971197</td>
<td>52</td>
<td>6658</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>21620</td>
<td>48473</td>
<td>17</td>
<td>6586</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>113491</td>
<td>384442</td>
<td>56</td>
<td>6616</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>22158</td>
<td>50872</td>
<td>19</td>
<td>2511</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>125478</td>
<td>440517</td>
<td>56</td>
<td>7095</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>16136</td>
<td>37944</td>
<td>16</td>
<td>32288</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>75959</td>
<td>265363</td>
<td>52</td>
<td>45923</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>13516</td>
<td>33683</td>
<td>18</td>
<td>29412</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>43201</td>
<td>150419</td>
<td>56</td>
<td>29428</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>6238</td>
<td>15285</td>
<td>27</td>
<td>15538</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>20178</td>
<td>69126</td>
<td>52</td>
<td>16467</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>14656</td>
<td>33223</td>
<td>12</td>
<td>3284</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>73964</td>
<td>267415</td>
<td>50</td>
<td>6055</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>3398</td>
<td>8342</td>
<td>26</td>
<td>8497</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>13525</td>
<td>41162</td>
<td>30</td>
<td>10250</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>5158</td>
<td>11729</td>
<td>9</td>
<td>2229</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>24749</td>
<td>84688</td>
<td>45</td>
<td>5274</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>1332</td>
<td>2878</td>
<td>6</td>
<td>661</td>
</tr>
<tr>
<td>21</td>
<td>2</td>
<td>6285</td>
<td>20378</td>
<td>26</td>
<td>3319</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>1523</td>
<td>4091</td>
<td>17</td>
<td>8515</td>
</tr>
<tr>
<td>22</td>
<td>2</td>
<td>3145</td>
<td>11408</td>
<td>36</td>
<td>9185</td>
</tr>
<tr>
<td>X</td>
<td>1</td>
<td>172484</td>
<td>379344</td>
<td>22</td>
<td>60635</td>
</tr>
<tr>
<td>X</td>
<td>2</td>
<td>1278415</td>
<td>4335081</td>
<td>131</td>
<td>60647</td>
</tr>
<tr>
<td>Y</td>
<td>1</td>
<td>5087</td>
<td>14054</td>
<td>21</td>
<td>57836</td>
</tr>
<tr>
<td>Y</td>
<td>2</td>
<td>27705</td>
<td>87337</td>
<td>40</td>
<td>57858</td>
</tr>
</tbody>
</table>

Tabelle C.14: Repeatstatistik zum native clustering von *homo sapiens* (unmaskiert)
<table>
<thead>
<tr>
<th>Chr.</th>
<th>PID</th>
<th>NUM_CLUSTERS</th>
<th>NUM_CL_PARTS</th>
<th>MAX_CL_PARTS</th>
<th>MAX_CL_LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>25787</td>
<td>54304</td>
<td>6</td>
<td>4574</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>235431</td>
<td>633757</td>
<td>24</td>
<td>4597</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>34255</td>
<td>72328</td>
<td>7</td>
<td>7131</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>333771</td>
<td>902971</td>
<td>24</td>
<td>7377</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>41370</td>
<td>87590</td>
<td>6</td>
<td>19328</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>388609</td>
<td>1046441</td>
<td>21</td>
<td>19343</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>32450</td>
<td>68740</td>
<td>7</td>
<td>4306</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>305688</td>
<td>816285</td>
<td>22</td>
<td>4497</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>24464</td>
<td>51572</td>
<td>8</td>
<td>13278</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>230873</td>
<td>615740</td>
<td>18</td>
<td>13297</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>17819</td>
<td>38055</td>
<td>8</td>
<td>9919</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>138621</td>
<td>377497</td>
<td>20</td>
<td>9949</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>19110</td>
<td>40611</td>
<td>10</td>
<td>13030</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>173067</td>
<td>469124</td>
<td>22</td>
<td>22026</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>10584</td>
<td>22395</td>
<td>5</td>
<td>3494</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>95909</td>
<td>258836</td>
<td>18</td>
<td>3197</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>12880</td>
<td>27073</td>
<td>7</td>
<td>4170</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>134447</td>
<td>358159</td>
<td>15</td>
<td>4194</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>7369</td>
<td>15646</td>
<td>6</td>
<td>9333</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>60353</td>
<td>162663</td>
<td>22</td>
<td>9364</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>8554</td>
<td>18079</td>
<td>6</td>
<td>2017</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>75420</td>
<td>199402</td>
<td>16</td>
<td>2035</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>7299</td>
<td>15477</td>
<td>6</td>
<td>10680</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>61770</td>
<td>165004</td>
<td>18</td>
<td>10722</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>13169</td>
<td>27914</td>
<td>7</td>
<td>6312</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>107110</td>
<td>285737</td>
<td>15</td>
<td>6328</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>4744</td>
<td>10074</td>
<td>7</td>
<td>4866</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>34588</td>
<td>93828</td>
<td>20</td>
<td>5042</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>5719</td>
<td>12150</td>
<td>6</td>
<td>8737</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>48259</td>
<td>128467</td>
<td>17</td>
<td>8761</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>3208</td>
<td>6851</td>
<td>8</td>
<td>4061</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>24598</td>
<td>66931</td>
<td>17</td>
<td>8799</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>2930</td>
<td>6242</td>
<td>6</td>
<td>7823</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>24279</td>
<td>65069</td>
<td>17</td>
<td>7835</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>1732</td>
<td>3766</td>
<td>8</td>
<td>20700</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>9910</td>
<td>26470</td>
<td>20</td>
<td>20714</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>706</td>
<td>1557</td>
<td>5</td>
<td>8163</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>2963</td>
<td>7588</td>
<td>13</td>
<td>8181</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>523</td>
<td>1116</td>
<td>5</td>
<td>1988</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>3766</td>
<td>9602</td>
<td>14</td>
<td>2012</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>1542</td>
<td>3338</td>
<td>6</td>
<td>2780</td>
</tr>
<tr>
<td>21</td>
<td>2</td>
<td>11010</td>
<td>29448</td>
<td>13</td>
<td>2801</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>435</td>
<td>954</td>
<td>5</td>
<td>6874</td>
</tr>
<tr>
<td>22</td>
<td>2</td>
<td>2585</td>
<td>6922</td>
<td>13</td>
<td>6943</td>
</tr>
<tr>
<td>23</td>
<td>1</td>
<td>242</td>
<td>514</td>
<td>4</td>
<td>2275</td>
</tr>
<tr>
<td>23</td>
<td>2</td>
<td>758</td>
<td>1868</td>
<td>8</td>
<td>3437</td>
</tr>
<tr>
<td>X</td>
<td>1</td>
<td>17313</td>
<td>30121</td>
<td>6</td>
<td>1488</td>
</tr>
<tr>
<td>X</td>
<td>2</td>
<td>207725</td>
<td>554495</td>
<td>19</td>
<td>2207</td>
</tr>
<tr>
<td>Y</td>
<td>1</td>
<td>71</td>
<td>154</td>
<td>5</td>
<td>1578</td>
</tr>
<tr>
<td>Y</td>
<td>2</td>
<td>566</td>
<td>1350</td>
<td>7</td>
<td>1843</td>
</tr>
</tbody>
</table>

Tabelle C.15: Repeatstatistik zum native clustering von *pan troglodytes* (unmaskiert)
<table>
<thead>
<tr>
<th>Chr</th>
<th>PID</th>
<th>NUM_CLUSTERS</th>
<th>NUM_CL_PARTS</th>
<th>MAX_CL_PARTS</th>
<th>MAX_CL_LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2177</td>
<td>5368</td>
<td>14</td>
<td>5203</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>1954</td>
<td>8024</td>
<td>23</td>
<td>5895</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>1005</td>
<td>2455</td>
<td>8</td>
<td>2122</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>1014</td>
<td>3385</td>
<td>20</td>
<td>2706</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>90</td>
<td>205</td>
<td>5</td>
<td>1391</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>248</td>
<td>820</td>
<td>18</td>
<td>2368</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>393</td>
<td>949</td>
<td>6</td>
<td>2647</td>
</tr>
<tr>
<td>4</td>
<td>2</td>
<td>622</td>
<td>1787</td>
<td>18</td>
<td>2730</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>991</td>
<td>2553</td>
<td>9</td>
<td>3707</td>
</tr>
<tr>
<td>5</td>
<td>2</td>
<td>1502</td>
<td>4974</td>
<td>24</td>
<td>4312</td>
</tr>
<tr>
<td>6</td>
<td>1</td>
<td>229</td>
<td>559</td>
<td>6</td>
<td>2400</td>
</tr>
<tr>
<td>6</td>
<td>2</td>
<td>379</td>
<td>1180</td>
<td>14</td>
<td>2417</td>
</tr>
<tr>
<td>7</td>
<td>1</td>
<td>959</td>
<td>2308</td>
<td>7</td>
<td>3047</td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>1263</td>
<td>3587</td>
<td>13</td>
<td>3341</td>
</tr>
<tr>
<td>8</td>
<td>1</td>
<td>1142</td>
<td>3186</td>
<td>10</td>
<td>4035</td>
</tr>
<tr>
<td>8</td>
<td>2</td>
<td>1006</td>
<td>4829</td>
<td>21</td>
<td>4053</td>
</tr>
<tr>
<td>9</td>
<td>1</td>
<td>1379</td>
<td>3578</td>
<td>11</td>
<td>2855</td>
</tr>
<tr>
<td>9</td>
<td>2</td>
<td>1002</td>
<td>3392</td>
<td>34</td>
<td>5460</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>1730</td>
<td>4447</td>
<td>15</td>
<td>6735</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>1228</td>
<td>4815</td>
<td>20</td>
<td>6781</td>
</tr>
<tr>
<td>11</td>
<td>1</td>
<td>189</td>
<td>433</td>
<td>5</td>
<td>2110</td>
</tr>
<tr>
<td>11</td>
<td>2</td>
<td>594</td>
<td>1905</td>
<td>11</td>
<td>2132</td>
</tr>
<tr>
<td>12</td>
<td>1</td>
<td>91</td>
<td>202</td>
<td>4</td>
<td>998</td>
</tr>
<tr>
<td>12</td>
<td>2</td>
<td>171</td>
<td>493</td>
<td>15</td>
<td>2271</td>
</tr>
<tr>
<td>13</td>
<td>1</td>
<td>60</td>
<td>139</td>
<td>6</td>
<td>768</td>
</tr>
<tr>
<td>13</td>
<td>2</td>
<td>79</td>
<td>225</td>
<td>8</td>
<td>1134</td>
</tr>
<tr>
<td>14</td>
<td>1</td>
<td>118</td>
<td>285</td>
<td>7</td>
<td>1354</td>
</tr>
<tr>
<td>14</td>
<td>2</td>
<td>182</td>
<td>654</td>
<td>13</td>
<td>1921</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1303</td>
<td>3365</td>
<td>11</td>
<td>4262</td>
</tr>
<tr>
<td>15</td>
<td>2</td>
<td>1299</td>
<td>4297</td>
<td>20</td>
<td>6176</td>
</tr>
<tr>
<td>16</td>
<td>1</td>
<td>2476</td>
<td>6046</td>
<td>10</td>
<td>5390</td>
</tr>
<tr>
<td>16</td>
<td>2</td>
<td>1948</td>
<td>6444</td>
<td>30</td>
<td>5406</td>
</tr>
<tr>
<td>17</td>
<td>1</td>
<td>972</td>
<td>2489</td>
<td>11</td>
<td>4526</td>
</tr>
<tr>
<td>17</td>
<td>2</td>
<td>853</td>
<td>3030</td>
<td>21</td>
<td>4539</td>
</tr>
<tr>
<td>18</td>
<td>1</td>
<td>18</td>
<td>41</td>
<td>3</td>
<td>2149</td>
</tr>
<tr>
<td>18</td>
<td>2</td>
<td>28</td>
<td>69</td>
<td>4</td>
<td>2166</td>
</tr>
<tr>
<td>19</td>
<td>1</td>
<td>387</td>
<td>931</td>
<td>5</td>
<td>1859</td>
</tr>
<tr>
<td>19</td>
<td>2</td>
<td>958</td>
<td>2843</td>
<td>16</td>
<td>3569</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>31</td>
<td>74</td>
<td>6</td>
<td>1557</td>
</tr>
<tr>
<td>20</td>
<td>2</td>
<td>91</td>
<td>295</td>
<td>10</td>
<td>1643</td>
</tr>
<tr>
<td>21</td>
<td>1</td>
<td>24</td>
<td>57</td>
<td>5</td>
<td>565</td>
</tr>
<tr>
<td>21</td>
<td>2</td>
<td>50</td>
<td>149</td>
<td>8</td>
<td>1034</td>
</tr>
<tr>
<td>22</td>
<td>1</td>
<td>566</td>
<td>1522</td>
<td>9</td>
<td>5028</td>
</tr>
<tr>
<td>22</td>
<td>2</td>
<td>430</td>
<td>1907</td>
<td>22</td>
<td>5480</td>
</tr>
<tr>
<td>X</td>
<td>1</td>
<td>422</td>
<td>1064</td>
<td>7</td>
<td>4404</td>
</tr>
<tr>
<td>X</td>
<td>2</td>
<td>773</td>
<td>2381</td>
<td>29</td>
<td>4422</td>
</tr>
<tr>
<td>Y</td>
<td>1</td>
<td>588</td>
<td>1651</td>
<td>9</td>
<td>4974</td>
</tr>
<tr>
<td>Y</td>
<td>2</td>
<td>660</td>
<td>2294</td>
<td>17</td>
<td>4990</td>
</tr>
</tbody>
</table>

Tabelle C.16: Repeatstatistik zum native clustering von *mus musculus* (unmaskiert)
<table>
<thead>
<tr>
<th>Chr.</th>
<th>PID</th>
<th>NUM_CLUSTERS</th>
<th>NUM_CL_PARTS</th>
<th>MAX_CL_PARTS</th>
<th>MAX_CL_LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>114312</td>
<td>257409</td>
<td>17</td>
<td>29817</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>676110</td>
<td>2315856</td>
<td>58</td>
<td>29826</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>155548</td>
<td>348905</td>
<td>17</td>
<td>100267</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>859481</td>
<td>3008849</td>
<td>59</td>
<td>100520</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>132668</td>
<td>294782</td>
<td>17</td>
<td>3315</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>829163</td>
<td>2893893</td>
<td>54</td>
<td>6038</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>149469</td>
<td>331692</td>
<td>19</td>
<td>9496</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>967211</td>
<td>3336596</td>
<td>62</td>
<td>9546</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>119053</td>
<td>265485</td>
<td>17</td>
<td>14680</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>792493</td>
<td>2725373</td>
<td>60</td>
<td>19874</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>83850</td>
<td>185050</td>
<td>20</td>
<td>5270</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>566489</td>
<td>1915194</td>
<td>56</td>
<td>14427</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>65014</td>
<td>145434</td>
<td>22</td>
<td>14898</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>360819</td>
<td>1249691</td>
<td>59</td>
<td>150582</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>66763</td>
<td>148878</td>
<td>15</td>
<td>11865</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>430697</td>
<td>1482921</td>
<td>55</td>
<td>14087</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>38293</td>
<td>87979</td>
<td>19</td>
<td>10055</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>230456</td>
<td>770747</td>
<td>49</td>
<td>15254</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>39616</td>
<td>88502</td>
<td>14</td>
<td>49885</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>240280</td>
<td>827055</td>
<td>50</td>
<td>50033</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>57567</td>
<td>126894</td>
<td>14</td>
<td>11199</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>390217</td>
<td>1304917</td>
<td>52</td>
<td>11211</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>45252</td>
<td>996692</td>
<td>17</td>
<td>2022</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>283431</td>
<td>966633</td>
<td>52</td>
<td>5974</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>21250</td>
<td>47131</td>
<td>10</td>
<td>6268</td>
</tr>
<tr>
<td>13</td>
<td>4</td>
<td>110562</td>
<td>375837</td>
<td>55</td>
<td>10540</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>21002</td>
<td>46259</td>
<td>14</td>
<td>3454</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>125217</td>
<td>435831</td>
<td>54</td>
<td>6032</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>13858</td>
<td>37531</td>
<td>16</td>
<td>12908</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>75691</td>
<td>262476</td>
<td>50</td>
<td>12920</td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td>13019</td>
<td>32960</td>
<td>22</td>
<td>49266</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>41314</td>
<td>144366</td>
<td>53</td>
<td>49282</td>
</tr>
<tr>
<td>17</td>
<td>3</td>
<td>5547</td>
<td>13447</td>
<td>19</td>
<td>16673</td>
</tr>
<tr>
<td>17</td>
<td>4</td>
<td>18772</td>
<td>62052</td>
<td>55</td>
<td>17959</td>
</tr>
<tr>
<td>18</td>
<td>3</td>
<td>14530</td>
<td>33051</td>
<td>11</td>
<td>3638</td>
</tr>
<tr>
<td>18</td>
<td>4</td>
<td>73468</td>
<td>264626</td>
<td>55</td>
<td>6054</td>
</tr>
<tr>
<td>19</td>
<td>3</td>
<td>2941</td>
<td>7018</td>
<td>9</td>
<td>4148</td>
</tr>
<tr>
<td>19</td>
<td>4</td>
<td>11824</td>
<td>36596</td>
<td>37</td>
<td>4165</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>4827</td>
<td>10946</td>
<td>11</td>
<td>20250</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>24579</td>
<td>83118</td>
<td>48</td>
<td>20271</td>
</tr>
<tr>
<td>21</td>
<td>3</td>
<td>1126</td>
<td>2443</td>
<td>5</td>
<td>784</td>
</tr>
<tr>
<td>21</td>
<td>4</td>
<td>5816</td>
<td>18516</td>
<td>24</td>
<td>2955</td>
</tr>
<tr>
<td>22</td>
<td>3</td>
<td>1495</td>
<td>4154</td>
<td>13</td>
<td>6934</td>
</tr>
<tr>
<td>22</td>
<td>4</td>
<td>3088</td>
<td>11571</td>
<td>32</td>
<td>10386</td>
</tr>
<tr>
<td>X</td>
<td>3</td>
<td>170451</td>
<td>375225</td>
<td>19</td>
<td>29654</td>
</tr>
<tr>
<td>X</td>
<td>4</td>
<td>1280951</td>
<td>4335519</td>
<td>59</td>
<td>74174</td>
</tr>
<tr>
<td>Y</td>
<td>3</td>
<td>3218</td>
<td>8562</td>
<td>22</td>
<td>54746</td>
</tr>
<tr>
<td>Y</td>
<td>4</td>
<td>24761</td>
<td>79318</td>
<td>38</td>
<td>54759</td>
</tr>
</tbody>
</table>

Tabelle C.17: Palindromstatistik zum native clustering von *homo sapiens* (unmaskiert)
<table>
<thead>
<tr>
<th>Chr.</th>
<th>PID</th>
<th>NUM CLUSTERS</th>
<th>NUM_CL_PARTS</th>
<th>MAX_CL_PARTS</th>
<th>MAX_CL_LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>24974</td>
<td>52504</td>
<td>8</td>
<td>1123</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>236278</td>
<td>637720</td>
<td>22</td>
<td>2568</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>33915</td>
<td>71604</td>
<td>7</td>
<td>1053</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>334743</td>
<td>905645</td>
<td>26</td>
<td>2420</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>40692</td>
<td>86201</td>
<td>7</td>
<td>1043</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>388847</td>
<td>1050006</td>
<td>24</td>
<td>2634</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>32326</td>
<td>68488</td>
<td>8</td>
<td>1295</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>306642</td>
<td>821647</td>
<td>21</td>
<td>2532</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>23895</td>
<td>50438</td>
<td>6</td>
<td>992</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>231680</td>
<td>618908</td>
<td>21</td>
<td>2457</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>17276</td>
<td>36799</td>
<td>9</td>
<td>1075</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>137818</td>
<td>374489</td>
<td>18</td>
<td>2201</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>19028</td>
<td>40353</td>
<td>7</td>
<td>1111</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>173744</td>
<td>471492</td>
<td>20</td>
<td>2432</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>10229</td>
<td>21538</td>
<td>7</td>
<td>1297</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>95266</td>
<td>256032</td>
<td>21</td>
<td>1985</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>12785</td>
<td>26837</td>
<td>9</td>
<td>1092</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>134085</td>
<td>355290</td>
<td>17</td>
<td>2348</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>7036</td>
<td>14879</td>
<td>6</td>
<td>724</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>60691</td>
<td>163132</td>
<td>18</td>
<td>2243</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>8475</td>
<td>17821</td>
<td>6</td>
<td>1215</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>75782</td>
<td>200949</td>
<td>20</td>
<td>2392</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>6788</td>
<td>14381</td>
<td>6</td>
<td>1490</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>60059</td>
<td>160909</td>
<td>19</td>
<td>2009</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>13165</td>
<td>27882</td>
<td>7</td>
<td>1092</td>
</tr>
<tr>
<td>13</td>
<td>4</td>
<td>104913</td>
<td>281090</td>
<td>17</td>
<td>1980</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>4622</td>
<td>9867</td>
<td>6</td>
<td>1015</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>34173</td>
<td>92830</td>
<td>20</td>
<td>2287</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>5580</td>
<td>11798</td>
<td>5</td>
<td>793</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>48257</td>
<td>128508</td>
<td>18</td>
<td>1983</td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td>3073</td>
<td>6563</td>
<td>6</td>
<td>1210</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>24638</td>
<td>67312</td>
<td>18</td>
<td>2211</td>
</tr>
<tr>
<td>17</td>
<td>3</td>
<td>2657</td>
<td>5612</td>
<td>6</td>
<td>1031</td>
</tr>
<tr>
<td>17</td>
<td>4</td>
<td>24051</td>
<td>64530</td>
<td>21</td>
<td>2418</td>
</tr>
<tr>
<td>18</td>
<td>3</td>
<td>1526</td>
<td>3268</td>
<td>6</td>
<td>1661</td>
</tr>
<tr>
<td>18</td>
<td>4</td>
<td>9768</td>
<td>26305</td>
<td>14</td>
<td>1894</td>
</tr>
<tr>
<td>19</td>
<td>3</td>
<td>541</td>
<td>1191</td>
<td>7</td>
<td>1366</td>
</tr>
<tr>
<td>19</td>
<td>4</td>
<td>2473</td>
<td>6371</td>
<td>12</td>
<td>1694</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>349</td>
<td>758</td>
<td>6</td>
<td>1223</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>3108</td>
<td>7857</td>
<td>10</td>
<td>1390</td>
</tr>
<tr>
<td>21</td>
<td>3</td>
<td>1445</td>
<td>3088</td>
<td>6</td>
<td>1062</td>
</tr>
<tr>
<td>21</td>
<td>4</td>
<td>10859</td>
<td>28790</td>
<td>12</td>
<td>1636</td>
</tr>
<tr>
<td>22</td>
<td>3</td>
<td>386</td>
<td>815</td>
<td>4</td>
<td>644</td>
</tr>
<tr>
<td>22</td>
<td>4</td>
<td>2448</td>
<td>6498</td>
<td>11</td>
<td>1295</td>
</tr>
<tr>
<td>23</td>
<td>3</td>
<td>143</td>
<td>308</td>
<td>6</td>
<td>772</td>
</tr>
<tr>
<td>23</td>
<td>4</td>
<td>719</td>
<td>1816</td>
<td>9</td>
<td>1327</td>
</tr>
<tr>
<td>X</td>
<td>3</td>
<td>17288</td>
<td>36124</td>
<td>6</td>
<td>932</td>
</tr>
<tr>
<td>X</td>
<td>4</td>
<td>206878</td>
<td>552872</td>
<td>19</td>
<td>4082</td>
</tr>
<tr>
<td>Y</td>
<td>3</td>
<td>86</td>
<td>197</td>
<td>6</td>
<td>2923</td>
</tr>
<tr>
<td>Y</td>
<td>4</td>
<td>572</td>
<td>1399</td>
<td>9</td>
<td>2944</td>
</tr>
</tbody>
</table>

Tabelle C.18: Palindromstatistik zum native clustering von *pan troglodytes* (unmaskiert)
<table>
<thead>
<tr>
<th>Chr.</th>
<th>PID</th>
<th>NUM_CLUSTERS</th>
<th>NUM_CL_PARTS</th>
<th>MAX_CL_PARTS</th>
<th>MAX_CL_LENGTH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>3</td>
<td>1757</td>
<td>4481</td>
<td>9</td>
<td>3274</td>
</tr>
<tr>
<td>1</td>
<td>4</td>
<td>1662</td>
<td>6535</td>
<td>20</td>
<td>3629</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>1762</td>
<td>4515</td>
<td>12</td>
<td>5831</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>1406</td>
<td>5006</td>
<td>21</td>
<td>5842</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>77</td>
<td>192</td>
<td>10</td>
<td>1941</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>191</td>
<td>646</td>
<td>17</td>
<td>2198</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>125</td>
<td>276</td>
<td>4</td>
<td>1120</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>356</td>
<td>1023</td>
<td>16</td>
<td>2679</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>1045</td>
<td>2701</td>
<td>11</td>
<td>3718</td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>1188</td>
<td>4010</td>
<td>23</td>
<td>4311</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>121</td>
<td>301</td>
<td>8</td>
<td>2943</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>205</td>
<td>631</td>
<td>15</td>
<td>2955</td>
</tr>
<tr>
<td>7</td>
<td>3</td>
<td>1514</td>
<td>3642</td>
<td>9</td>
<td>4807</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>1450</td>
<td>4526</td>
<td>22</td>
<td>5592</td>
</tr>
<tr>
<td>8</td>
<td>3</td>
<td>1260</td>
<td>3487</td>
<td>9</td>
<td>3772</td>
</tr>
<tr>
<td>8</td>
<td>4</td>
<td>1071</td>
<td>5535</td>
<td>24</td>
<td>3789</td>
</tr>
<tr>
<td>9</td>
<td>3</td>
<td>1805</td>
<td>4672</td>
<td>9</td>
<td>3853</td>
</tr>
<tr>
<td>9</td>
<td>4</td>
<td>1201</td>
<td>4073</td>
<td>26</td>
<td>4073</td>
</tr>
<tr>
<td>10</td>
<td>3</td>
<td>1013</td>
<td>2585</td>
<td>9</td>
<td>2561</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>808</td>
<td>3129</td>
<td>20</td>
<td>3173</td>
</tr>
<tr>
<td>11</td>
<td>3</td>
<td>431</td>
<td>999</td>
<td>7</td>
<td>2937</td>
</tr>
<tr>
<td>11</td>
<td>4</td>
<td>973</td>
<td>3176</td>
<td>12</td>
<td>2984</td>
</tr>
<tr>
<td>12</td>
<td>3</td>
<td>162</td>
<td>429</td>
<td>8</td>
<td>1701</td>
</tr>
<tr>
<td>12</td>
<td>4</td>
<td>141</td>
<td>462</td>
<td>9</td>
<td>2002</td>
</tr>
<tr>
<td>13</td>
<td>3</td>
<td>90</td>
<td>218</td>
<td>5</td>
<td>1542</td>
</tr>
<tr>
<td>13</td>
<td>4</td>
<td>109</td>
<td>325</td>
<td>11</td>
<td>1616</td>
</tr>
<tr>
<td>14</td>
<td>3</td>
<td>20</td>
<td>56</td>
<td>6</td>
<td>1294</td>
</tr>
<tr>
<td>14</td>
<td>4</td>
<td>30</td>
<td>141</td>
<td>15</td>
<td>1994</td>
</tr>
<tr>
<td>15</td>
<td>3</td>
<td>1616</td>
<td>4176</td>
<td>10</td>
<td>4756</td>
</tr>
<tr>
<td>15</td>
<td>4</td>
<td>1467</td>
<td>4850</td>
<td>21</td>
<td>4769</td>
</tr>
<tr>
<td>16</td>
<td>3</td>
<td>2359</td>
<td>5907</td>
<td>15</td>
<td>5242</td>
</tr>
<tr>
<td>16</td>
<td>4</td>
<td>1738</td>
<td>5662</td>
<td>36</td>
<td>5257</td>
</tr>
<tr>
<td>17</td>
<td>3</td>
<td>715</td>
<td>1824</td>
<td>10</td>
<td>4524</td>
</tr>
<tr>
<td>17</td>
<td>4</td>
<td>670</td>
<td>2174</td>
<td>20</td>
<td>4547</td>
</tr>
<tr>
<td>18</td>
<td>3</td>
<td>45</td>
<td>113</td>
<td>6</td>
<td>1072</td>
</tr>
<tr>
<td>18</td>
<td>4</td>
<td>26</td>
<td>93</td>
<td>14</td>
<td>2344</td>
</tr>
<tr>
<td>19</td>
<td>3</td>
<td>324</td>
<td>935</td>
<td>6</td>
<td>1110</td>
</tr>
<tr>
<td>19</td>
<td>4</td>
<td>312</td>
<td>1049</td>
<td>8</td>
<td>1954</td>
</tr>
<tr>
<td>20</td>
<td>3</td>
<td>38</td>
<td>89</td>
<td>4</td>
<td>713</td>
</tr>
<tr>
<td>20</td>
<td>4</td>
<td>47</td>
<td>149</td>
<td>13</td>
<td>1596</td>
</tr>
<tr>
<td>21</td>
<td>3</td>
<td>6</td>
<td>12</td>
<td>2</td>
<td>628</td>
</tr>
<tr>
<td>21</td>
<td>4</td>
<td>15</td>
<td>42</td>
<td>5</td>
<td>832</td>
</tr>
<tr>
<td>22</td>
<td>3</td>
<td>620</td>
<td>1656</td>
<td>9</td>
<td>4316</td>
</tr>
<tr>
<td>22</td>
<td>4</td>
<td>502</td>
<td>2145</td>
<td>21</td>
<td>5436</td>
</tr>
<tr>
<td>X</td>
<td>3</td>
<td>246</td>
<td>578</td>
<td>6</td>
<td>4569</td>
</tr>
<tr>
<td>X</td>
<td>4</td>
<td>561</td>
<td>1587</td>
<td>15</td>
<td>5891</td>
</tr>
<tr>
<td>Y</td>
<td>3</td>
<td>511</td>
<td>1310</td>
<td>9</td>
<td>6029</td>
</tr>
<tr>
<td>Y</td>
<td>4</td>
<td>664</td>
<td>1942</td>
<td>24</td>
<td>6047</td>
</tr>
</tbody>
</table>

Tabelle C.19: Palindromstatistik zum native clustering von *mus musculus* (unmaskiert)
Abbildung D.1: Anzahl berechneter Repeats und Palindrome im Schimpanzengenom (V. 23)
Abbildung D.2: Anzahl berechneter Repeats und Palindrome im Mausgenom

D.1 Abbildungen zur durchschnittlichen Duplikatarmlänge in diversen Topologien des unmaskierten Humangenoms
Abbildung D.3: $\emptyset$ Palindromarmlängen

Abbildung D.4: $\emptyset$ native Palindromarmlängen
Abbildung D.5: ∅ Palindromarmlängen

Abbildung D.6: ∅ Repeatarmlängen
Abbildung D.7: $\emptyset$ Repeatarmlängen

Abbildung D.8: $\emptyset$ native Repeatarmlängen
D.2 Abbildungen zur prozentualen Duplikatverteilung in diversen Topologien des unmaskierten Human-genoms
Abbildung D.10: Proz. Aufteilung der Palindrome

Abbildung D.11: Proz. Aufteilung der native Palindrome
Abbildung D.12: Proz. Aufteilung der native Palindrome

Abbildung D.13: Proz. Aufteilung der Repeats
Abbildung D.14: Proz. Aufteilung der Repeats

Abbildung D.15: Proz. Aufteilung der native Repeats
Abbildung D.16: Proz. Aufteilung der native Repeats